



ESCUELA SUPERIOR DE INGENIERÍA

INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS

Sistema de monitorización de desarrollo de
software colaborativo mediante análisis de actividad
de repositorios Subversion y blogs

Rubén Díaz Sánchez

Cádiz, Septiembre 2011



ESCUELA SUPERIOR DE INGENIERÍA

INGENIERO TÉCNICO EN INFORMÁTICA DE SISTEMAS

- Departamento: Lenguajes y Sistemas Informáticos
- Director del proyecto: Manuel Palomo Duarte
- Codirector del proyecto: Antonio Balderas Alberico
- Autor del proyecto: Rubén Díaz Sánchez

Cádiz, Septiembre 2011
Fdo: Rubén Díaz Sánchez

Licencia

Este documento ha sido liberado bajo Licencia GFDL 1.3 (GNU Free Documentation License). Se incluyen los términos de la licencia en inglés al final del mismo.

Copyright (c) 2011 Rubén Díaz Sánchez.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Agradecimientos

Todo este trabajo se lo dedico a mi familia, en concreto a mi madre, que a cada momento se ha paseado a ver si estaba trabajando. Quiero agradecer el trabajo de mi tutor (Manuel Palomo) y a Antonio Balderas por toda la ayuda que me han prestado. También se lo dedico, por supuesto, a los compañeros que estuvieron, a los que están y a los que estarán pasando conmigo esta época de mi vida tan importante. En concreto a Charly, Javi, Marcos, Antonio, Pepelu, Ale porque han hecho que todo este tiempo de trabajo haya sido agradable, ameno y divertido. No puedo dejar pasar a mis grandes amigos Ismael y Sara, que a pesar de no tener ni idea de como va todo esto, se han dejado sus oídos y su corazón en intentar entenderme y a mi suegro Paco, que me ha dado tantas ideas y consejos para seguir avanzando. Para terminar dejo a la más valiente de todas, mi pareja Mely, que me ha aguantado día tras día, con el proyecto en la cabeza. Que me ha comprendido cuando en mitad de una conversación desaparecía para repasar los entresijos del código. Y que me ha dado todo el apoyo que necesitaba en los momentos más y menos difíciles.

Por todo esto y mucho más, gracias a todos por ayudarme a andar este camino.

Índice general

Licencia	III
Agradecimientos	V
1. Introducción	1
1.1. Motivación y Justificación	1
1.2. Objetivos y Alcance	2
1.3. Definiciones, acrónimos y abreviaturas	3
1.4. Equipo de Desarrollo	6
1.5. Visión General	6
2. Desarrollo del Calendario	9
2.1. Actividades Mensuales	9
2.1.1. Diagrama de Gantt	12
2.2. Crecimiento del Diagrama de Casos de Uso	14
2.3. Horas Dedicadas	15
3. Descripción General	17
3.1. Perspectivas del Producto	17
3.2. Funciones del Producto	19
3.3. Características del Usuario	20
3.4. Restricciones Generales	20
3.5. Suposiciones y Dependencias	21
3.6. Requisitos para Futuras Versiones del Sistema	21

4. Especificación de Requisitos	23
4.1. Introducción	23
4.2. Metodología de Desarrollo	23
4.3. Requisitos Específicos	25
4.3.1. Requisitos de Interfaces Externos	25
4.3.2. Requisitos Funcionales	26
4.3.3. Requisitos de Rendimiento	29
4.3.4. Otros Requisitos	30
5. Análisis del Sistema	31
5.1. Introducción	31
5.2. Diagramas de Casos de Uso	31
5.2.1. Diagrama de Casos de Uso: Gestión de Alumnos	32
5.2.2. Diagrama de Casos de Uso: Gestión de Proyectos	33
5.2.3. Diagrama de Casos de Uso: Gestión de Profesores	34
5.2.4. Diagrama de Casos de Uso: Gestión del Sistema	35
5.3. Especificación de Casos de Uso	36
5.3.1. Especificación del Caso de Uso: Alta Alumno	36
5.3.2. Especificación del Caso de Uso: Listar Alumnos	37
5.3.3. Especificación del Caso de Uso: Listar todos los Alumnos	37
5.3.4. Especificación del Caso de Uso: Borrar Alumno	38
5.3.5. Especificación del Caso de Uso: Vincular Proyecto	38
5.3.6. Especificación del Caso de Uso: Vincular Desarrollador	39
5.3.7. Especificación del Caso de Uso: Mostrar Estadísticas según Profesor	40
5.3.8. Especificación del Caso de Uso: Mostrar Estadísticas según Proyecto	40
5.3.9. Especificación del Caso de Uso: Login Alumno	41

5.3.10. Especificación del Caso de Uso: Alta Proyecto	42
5.3.11. Especificación del Caso de Uso: Listar Proyectos	43
5.3.12. Especificación del Caso de Uso: Listar todos los Proyectos	43
5.3.13. Especificación del Caso de Uso: Borrar Proyecto	44
5.3.14. Especificación del Caso de Uso: Mostrar Estadísticas (Proy)	44
5.3.15. Especificación del Caso de Uso: Actualizar Proyecto	45
5.3.16. Especificación del Caso de Uso: Descargar CSV	46
5.3.17. Especificación del Caso de Uso: Alta Profesor	47
5.3.18. Especificación del Caso de Uso: Listar Profesores	48
5.3.19. Especificación del Caso de Uso: Borrar Profesor	49
5.3.20. Especificación del Caso de Uso: Cambiar Contraseña	49
5.3.21. Especificación del Caso de Uso: Configurar Nivel Proyectos	50
5.3.22. Especificación del Caso de Uso: Configurar Nivel Alumnos	51
5.3.23. Especificación del Caso de Uso: Login Profesor	51
5.3.24. Especificación del Caso de Uso: Actualizar Todo	52
5.3.25. Especificación del Caso de Uso: Configurar Actualización	53
5.3.26. Especificación del Caso de Uso: Listar Cuentas Nuevas	53
5.3.27. Especificación del Caso de Uso: Listar Copias de Seguridad	54
5.3.28. Especificación del Caso de Uso: Crear Copia de Seguridad	54
5.3.29. Especificación del Caso de Uso: Borrar Copia de Seguridad	55
5.3.30. Especificación del Caso de Uso: Descargar Copia de Seguridad	55
5.3.31. Especificación del Caso de Uso: Login Administrador	56
5.3.32. Especificación del Caso de Uso: Cambiar Contraseña Administrador	56
5.3.33. Especificación del Caso de Uso: Cerrar Sesión	57
5.4. Modelo Conceptual de Datos	58

5.4.1.	Entidades y Atributos	58
5.4.2.	Diagrama de Clases Conceptuales	59
5.4.3.	Restricciones y Reglas	60
5.5.	Modelo de Comportamiento del Sistema	60
5.5.1.	Diagramas de Secuencia del Sistema	60
5.5.2.	Contratos de las Operaciones del Sistema	74
6.	Diseño del Sistema	83
6.1.	Introducción	83
6.2.	Arquitectura Modelo-Vista-Controlador	83
6.3.	Capa de Vistas	85
6.4.	Capa de Controladores (<i>Diagramas de Interacción</i>)	85
6.4.1.	Diagramas de Interacción: Sección de Alumnos	85
6.4.2.	Diagrama de Interacción: Actualizar Proyecto	89
6.4.3.	Diagrama de Interacción: Alta Alumno	89
6.4.4.	Diagrama de Interacción: Mostrar Estadísticas de Proyecto	90
6.4.5.	Diagrama de Interacción: Borrar Proyecto	90
6.4.6.	Diagrama de Interacción: Configurar Nivel Proyectos	91
6.4.7.	Diagrama de Interacción: Cerrar Sesión	91
6.4.8.	Diagrama de Interacción: Crear Copia de Seguridad	92
6.4.9.	Diagrama de Interacción: Descargar CSV	92
6.4.10.	Diagrama de Interacción: Mostrar Estadísticas de Proyecto	93
6.4.11.	Diagrama de Interacción: Login Profesor	93
6.4.12.	Diagrama de Interacción: Vincular Proyecto	94
6.4.13.	Contrato de las Operaciones	94
6.5.	Capa de Modelos	102
6.5.1.	Descripción	102
6.5.2.	Esquema Relacional	102

<i>ÍNDICE GENERAL</i>	XI
7. Codificación	105
7.1. Tecnologías	105
7.1.1. Nivel de Presentación (Vistas)	105
7.1.2. Nivel de Aplicación (Controladores y Modelos)	106
7.1.3. Nivel de Persistencia (Modelos)	108
7.2. Herramientas	109
7.2.1. Apache	109
7.2.2. Gimp	110
7.2.3. Gedit	110
7.2.4. NetBeans	111
7.2.5. Gantt Project	112
7.2.6. Forja RedIris	112
8. Pruebas	113
8.1. Tipos de Pruebas	113
8.1.1. Pruebas de Validación y Verificación	113
8.1.2. Pruebas de Integración	113
8.2. Conjuntos de Entrada	114
8.2.1. Pruebas durante el Desarrollo	114
8.2.2. Pruebas Finales	114
9. Conclusiones	115
9.1. Valoración Personal	115
9.2. Futuras Ampliaciones	116

A. Manual de Usuario	117
A.1. Sección para Alumnos	118
A.1.1. Login Alumno	118
A.1.2. Mostrar Estadísticas según Proyecto	119
A.1.3. Mostrar Estadísticas según Profesor	120
A.2. Sección para Profesores	121
A.2.1. Login Profesor	121
A.2.2. Alta Alumno	122
A.2.3. Alta Proyecto	123
A.2.4. Listar Alumnos	124
A.2.5. Listar Proyectos	125
A.2.6. Configurar Nivel Proyectos	126
A.2.7. Configurar Nivel Alumnos	127
A.2.8. Cambiar Contraseña	129
A.2.9. Vincular Proyecto	131
A.2.10. Vincular Desarrollador	132
A.2.11. Borrar Alumno	133
A.2.12. Borrar Proyecto	134
A.2.13. Mostrar Estadísticas (Proy)	136
A.2.14. Actualizar Proyecto	138
A.2.15. Descargar CSV	140
A.2.16. Alta Profesor	141
A.3. Sección para Administrador	143
A.3.1. Login Administrador	143
A.3.2. Listar Profesores	144

A.3.3. Listar Todos los Proyectos	145
A.3.4. Listar Todos los Alumnos	146
A.3.5. Listar Nuevas Cuentas	147
A.3.6. Lista de Copias de Seguridad	148
A.3.7. Actualizar Todo	149
A.3.8. Cambiar Contraseña de Administrador	150
A.3.9. Cambiar Configuración de actualización	152
A.4. Cerrar Sesión	153
A.4.1. Cerrar Sesión de Alumno	153
A.4.2. Cerrar Sesión de Profesor	154
A.4.3. Cerrar Sesión de Administrador	155
B. Manual de Instalación	157
B.1. Introducción	157
B.2. El Servidor	157
B.2.1. Instalación del Servidor Apache	158
B.2.2. Instalación del intérprete de PHP	158
B.2.3. Instalación de la Base de Datos MySQL	159
B.3. Instalando Librerías Necesarias	160
B.3.1. Instalar la librería GD	160
B.3.2. Instalar la librería SVN	160
B.4. Comprobando el Funcionamiento	161
B.5. Instalar la Aplicación	162
B.5.1. Creando la Base de Datos	162
B.5.2. Copiando los Archivos	162
B.5.3. Otorgando Permisos	163
B.5.4. Iniciando la Aplicación	163

Anexos	165
Bibliografía	167
GNU Free Documentation License	169
1. APPLICABILITY AND DEFINITIONS	169
2. VERBATIM COPYING	171
3. COPYING IN QUANTITY	171
4. MODIFICATIONS	172
5. COMBINING DOCUMENTS	174
6. COLLECTIONS OF DOCUMENTS	174
7. AGGREGATION WITH INDEPENDENT WORKS	175
8. TRANSLATION	175
9. TERMINATION	175
10. FUTURE REVISIONS OF THIS LICENSE	176
11. RELICENSING	176
ADDENDUM: How to use this License for your documents	177

Índice de Figuras

2.1. Diagrama de Gantt (1)	12
2.2. Diagrama de Gantt (2)	13
2.3. Crecimiento de los casos de uso.	14
2.4. Horas Mensuales.	15
5.1. Diagrama de Casos de Uso: Gestión de Alumnos.	32
5.2. Diagrama de Casos de Uso: Gestión de Proyectos.	33
5.3. Diagrama de Casos de Uso: Gestión de Profesores.	34
5.4. Diagrama de Casos de Uso: Gestión del Sistema.	35
5.5. Diagrama de Clases Conceptuales.	59
5.6. Diagrama de Secuencia del Sistema: Alta Alumno.	61
5.7. Diagrama de Secuencia del Sistema: Borrar Alumno.	61
5.8. Diagrama de Secuencia del Sistema: Vincular Proyecto.	62
5.9. Diagrama de Secuencia del Sistema: Vincular Desarrollador.	62
5.10. Diagrama de Secuencia del Sistema: Mostrar Estadísticas según Profesor. .	63
5.11. Diagrama de Secuencia del Sistema: Mostrar Estadísticas según Proyecto. .	63
5.12. Diagrama de Secuencia del Sistema: Login Alumno.	64
5.13. Diagrama de Secuencia del Sistema: Alta Proyecto.	64
5.14. Diagrama de Secuencia del Sistema: Borrar Proyecto.	65

5.15. Diagrama de Secuencia del Sistema: Mostrar Estadísticas (Proy).	65
5.16. Diagrama de Secuencia del Sistema: Actualizar Proyecto.	66
5.17. Diagrama de Secuencia del Sistema: Descargar CSV.	66
5.18. Diagrama de Secuencia del Sistema: Alta Profesor.	67
5.19. Diagrama de Secuencia del Sistema: Borrar Profesor.	67
5.20. Diagrama de Secuencia del Sistema: Cambiar Contraseña.	68
5.21. Diagrama de Secuencia del Sistema: Configurar Nivel Proyectos.	68
5.22. Diagrama de Secuencia del Sistema: Configurar Nivel Alumnos.	69
5.23. Diagrama de Secuencia del Sistema: Login Profesor.	69
5.24. Diagrama de Secuencia del Sistema: Actualizar Todo.	70
5.25. Diagrama de Secuencia del Sistema: Configurar Actualización.	70
5.26. Diagrama de Secuencia del Sistema: Crear Copia de Seguridad.	71
5.27. Diagrama de Secuencia del Sistema: Borrar Copia de Seguridad.	71
5.28. Diagrama de Secuencia del Sistema: Descargar Copia de Seguridad.	72
5.29. Diagrama de Secuencia del Sistema: Login Administrador.	72
5.30. Diagrama de Secuencia del Sistema: Cambiar Contraseña Administrador.	73
5.31. Diagrama de Secuencia del Sistema: Cerrar Sesión.	73
6.1. Arquitectura Modelo-Vista-Controlador.	84
6.2. Diagrama de Interacción del Menú Principal de Administrador.	88
6.3. Diagrama de Interacción del Caso de Uso: Actualizar Proyecto.	89
6.4. Diagrama de Interacción del Caso de Uso: Alta Alumno.	89
6.5. Diagrama de Interacción del Caso de Uso: Mostrar Estadísticas de Proyecto.	90
6.6. Diagrama de Interacción del Caso de Uso: Borrar Proyecto.	90
6.7. Diagrama de Interacción del Caso de Uso: Configurar Nivel Proyectos.	91
6.8. Diagrama de Interacción del Caso de Uso: Cerrar Sesión.	91

6.9. Diagrama de Interacción del Caso de Uso: Crear Copia de Seguridad.	92
6.10. Diagrama de Interacción del Caso de Uso: Descargar CSV.	92
6.11. Diagrama de Interacción del Caso de Uso: Mostrar Estadísticas de Proyecto.	93
6.12. Diagrama de Interacción del Caso de Uso: Login Profesor.	93
6.13. Diagrama de Interacción del Caso de Uso: Vincular Proyecto.	94
A.1. Pantalla Inicial	118
A.2. Pantalla Login Alumno	118
A.3. Pantalla Éxito Login Alumno	118
A.4. Pantalla Lista Proyectos	119
A.5. Pantalla Estadísticas de Alumno según Proyecto	119
A.6. Pantalla selección de opción	120
A.7. Pantalla Estadísticas de Alumno según Profesor	120
A.8. Pantalla Inicial	121
A.9. Pantalla Login Profesor	121
A.10. Pantalla Alta Alumno	122
A.11. Pantalla Alumno Creado	122
A.12. Pantalla Alta Proyecto	123
A.13. Pantalla Proyecto Creado	123
A.14. Pantalla Lista Alumnos	124
A.15. Pantalla Lista Proyectos	125
A.16. Pantalla de Configuración	126
A.17. Pantalla Configuración Actualizada	126
A.18. Pantalla de Configuración	127
A.19. Pantalla Cambiar Tipo de Configuración	127
A.20. Pantalla Configuración de Alumnos	128

A.21. Pantalla Configuración Actualizada	128
A.22. Pantalla de Configuración	129
A.23. Pantalla Cambiar Tipo de Configuración	129
A.24. Pantalla Configurar Contraseña	130
A.25. Pantalla Contraseña Actualizada	130
A.26. Pantalla Lista Alumnos	131
A.27. Pantalla Lista Proyectos	131
A.28. Pantalla Lista Alumnos	132
A.29. Pantalla Lista Desarrolladores	132
A.30. Pantalla Lista Alumnos	133
A.31. Pantalla Confirmar	133
A.32. Pantalla Lista Proyectos	134
A.33. Pantalla Gestionar	134
A.34. Pantalla Confirmar Borrar proyecto	135
A.35. Pantalla Éxito Borrar proyecto	135
A.36. Pantalla Lista Proyectos	136
A.37. Pantalla Gestionar	136
A.38. Pantalla Mostrar Estadísticas de Proyecto	137
A.39. Pantalla Lista Proyectos	138
A.40. Pantalla Gestionar	138
A.41. Pantalla Actualizar Proyecto	139
A.42. Pantalla Lista Proyectos	140
A.43. Pantalla Gestionar	140
A.44. Pantalla Inicio	141
A.45. Pantalla Gestionar	141

A.46. Pantalla Gestionar	142
A.47. Pantalla Login Administrador	143
A.48. Pantalla Login Administrador	143
A.49. Pantalla Lista Profesores	144
A.50. Pantalla Lista Proyectos	145
A.51. Pantalla Lista Alumnos	146
A.52. Pantalla Lista de Nuevas Cuentas	147
A.53. Pantalla Lista de Copias de Seguridad	148
A.54. Pantalla Actualizar Todo	149
A.55. Pantalla Actualizar Todo	149
A.56. Pantalla Actualizar Todo	149
A.57. Pantalla de Configuración	150
A.58. Pantalla Cambiar Tipo de Configuración	150
A.59. Pantalla Configurar Contraseña	151
A.60. Pantalla Contraseña Actualizada	151
A.61. Pantalla de Configuración	152
A.62. Pantalla Contraseña Actualizada	152
A.63. Pantalla de Cerrar Sesión	153
A.64. Pantalla de Cerrar Sesión	154
A.65. Pantalla de Cerrar Sesión	155

Índice de Tablas

1.1. <i>Equipo Hardware empleado para el proyecto.</i>	6
1.2. <i>Equipo Software empleado para el proyecto.</i>	6
6.1. Opciones del Menú de Alumnos	85
6.2. Opciones del Menú Alumnos-Proyectos	85
6.3. Opciones del Menú Alumno-Proyectos-Gestionar	86
6.4. Opciones del Menú de Profesor	86
6.5. Opciones del Menú Profesor-Mostrar Alumnos	86
6.6. Opciones del Menú Profesor-Mostrar Proyectos-Gestionar	87
6.7. Opciones del Menú Profesor-Configuración	87
6.8. Opciones del Menú de Profesor	87
6.9. Opciones del Menú de Profesor	87
6.10. Opciones del Menú de Profesor	87

Capítulo 1

Introducción

Este documento pone fin a una etapa de trabajo. Una etapa en la que hemos buscado desarrollar una aplicación capaz de seguir el trabajo de los desarrolladores en un proyecto, con la meta de poder observar la evolución y los cambios que sufre un proyecto a lo largo de su vida y, de igual forma, la evolución de los participantes que trabajan en él.

Con los capítulos que se irán desarrollando a lo largo de las siguientes páginas se explicará ese trabajo. En qué ha consistido, problemas, soluciones y muchas otras cosas que intentarán plasmar el proyecto que he desarrollado y cómo lo he hecho.

Este ha sido un proyecto real, yo he cumplido mi papel de Analista, Ingeniero de Requisitos, Desarrollador ... y Manuel Palomo ha sido mi cliente. Ya que esta aplicación será para su uso académico, este proyecto ha hecho que cumpla un papel real y por lo tanto, me ha permitido aprender mucho de errores que he ido cometiendo en el camino.

1.1. Motivación y Justificación

Cuando llegó la hora de empezar a trabajar en mi Proyecto Fin de Carrera hablé con compañeros que ya habían desarrollado o estaban trabajando en su PFC. Además de la buena crítica que recibía el señor Manuel Palomo, yo ya le conocía y había cursado la asignatura de Diseño de Videojuegos. Por lo que tras hablar con él algunas veces me propuso este proyecto. Nunca había urgado en programación web y me interesaba mucho aprender a hacerlo, además de hacerla trabajar con software externo (Subversion en este caso).

Me gustó la idea y decidí aceptar este PFC. Después de esto, se dieron las primeras reuniones en las que Manuel y yo hablamos de cómo sería el proyecto, qué incluiría, qué lenguaje de programación usar. Se trató sobre todo la idea principal, que fue crear una aplicación web capaz de controlar el trabajo de los desarrolladores en un proyecto, bastante interesante a nivel académico y sobre todo, de la asignatura de Diseño de Videojuegos.

1.2. Objetivos y Alcance

Este proyecto intenta cumplir un gran objetivo dividida en varias partes:

- El gran objetivo es crear una aplicación web capaz de ofrecer, de forma sencilla, la posibilidad de observar el desarrollo de cada persona dentro de un proyecto. Todo esto se consigue a través del desarrollo colaborativo con Subversion, con la dificultad que conlleva conseguir una correcta comunicación entre sistemas. Esta aplicación tiene grandes usos en fines docentes, permitiendo hacer un seguimiento del trabajo del alumno en un proyecto, cosa que normalmente no es posible, haciendo imposible de controlar los verdaderos trabajadores de un proyecto en grupo.

Para esto necesitamos otros objetivos:

1. **Mantener una Base de Datos actualizada con toda la información relevante para el sistema** (alumnos, profesores, proyectos). Esto conlleva una zona de gestión de información (alta, baja, modificaciones).
2. **Monitorizar la información** del repositorio **Subversion** asociado a los Proyectos.
3. **Dar la posibilidad a los alumnos de ver su trabajo**. Para lo cual, existirá una opción especial de acceso.
4. **Administrar la información** desde una zona de administración.
5. **Interfaz Sencillo** para un uso agradable y cómodo.

En cuestiones académicas los objetivos del proyecto consisten en la obtención de conocimientos referentes al desarrollo de aplicaciones de gestión de información. En concreto las siguientes aptitudes:

1. Iniciarse en la programación con los lenguajes *HTML*, *CSS*, *PHP*, *CodeIgniter*.
2. Aprender cómo funciona *Subversion* y cómo guarda la información.
3. Tomar un mayor contacto con *Bases de Datos MySQL*.
4. Familiarizarse con el uso del *servidor web Apache*.
5. Aprender a hacer funcionar un servidor apache y poder ejecutar la aplicación.
6. Breve iniciación a la *programación con JavaScript*.
7. Obtener las destrezas para obtener la información del cliente.

Como nombre identificativo del proyecto he usado el acrónimo **estDV**. Viene de la idea principal del proyecto, crear una aplicación web capaz de obtener información estadística del desarrollo de un Videojuego: estadística Diseño de Videojuegos. Aunque al final se expandió para cualquier tipo de proyecto alojado en un repositorio subversion, el nombre se quedó como identificativo. Así que podremos hacer referencia al proyecto como estDV.

En cuanto a lo que hace y no hace el producto, podemos decir que el producto está creado para controlar el trabajo de alumnos en un proyecto. Es decir, entre las personas que usen este proyecto, habrá quienes cumplan el rol de Profesor, los cuales tendrán asociados a otras personas con el rol de Alumno y Desarrollador. Los Profesores serán los encargados de dar alta a alumnos y proyectos, y los alumnos simplemente habrán de trabajar en su proyecto, sin tener más preocupación en estDV. Por último habrá un último rol, el Administrador, que se encargará del buen funcionamiento de este proyecto.

El proyecto es bastante genérico y amoldable a las necesidades de distintos propósitos que pudieran tener distintos Profesores. Además es muy escalable. Se le pueden añadir funcionalidades al proyecto sin mayor problema para ajustarse a nuevas necesidades que puedan surgir.

Este proyecto ha sido creado pensando en fines docentes, sin embargo, no es su única posibilidad de uso. Cualquier empresa podría usar esta aplicación en el ámbito profesional para monitorizar el trabajo de sus empleados. Además de contar con la monitorización de los blogs que hace ir un paso más allá en la observación del trabajo personal.

La principal innovación que se pretende incluir en este proyecto es la automatización de la información. De forma que con solo un vistazo rápido se pueda observar el progreso del proyecto en todas sus facetas. Esta comodidad va acompañada de una sencillez visual y sobretodo la sencillez en el uso de la aplicación, ya que sin esto, rompería la esencia del proyecto haciendolo inservible.

Un aspecto importante de este proyecto es que ha sido desarrollado en un entorno de desarrollo muy eficiente como es *CodeIgniter* de *EllisLab*. Es un entorno de desarrollo abierto que permite crear webs dinámicas con PHP. Su principal objetivo es ayudar a que los desarrolladores, puedan realizar proyectos mucho más rápido que creando toda la estructura desde cero.

En resumen, este trabajo, pretende proporcionar una herramienta potente a la vez que sencilla para ayudar al seguimiento de los proyectos de la asignatura de Diseño de Videojuegos de la UCA.

1.3. Definiciones, acrónimos y abreviaturas

Alojamiento Web (Web Hosting): es el servicio que provee a los usuarios de Internet un sistema para poder almacenar información, imágenes, vídeo, o cualquier contenido

accesible vía Web. Los Web Host son compañías que proporcionan espacio de un servidor a sus clientes.

API (Application Program Interface): representa una interfaz de comunicación entre componentes software. Se trata del conjunto de llamadas a ciertas bibliotecas que ofrecen acceso a ciertos servicios desde los procesos y representa un método para conseguir abstracción en la programación, generalmente (aunque no necesariamente) entre los niveles o capas inferiores y los superiores del software.

Apache, Servidor: es un software servidor HTTP de código abierto para plataformas Unix, GNU/Linux, Windows, Macintosh, y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual, que se desarrolla dentro del proyecto HTTP Server (HTTPD) de la Apache Software Foundation.

CSS (Cascade Style Sheet): las Hojas de Estilo en Cascada o CSS constituyen el estándar para la inserción de estilos (tamaños, colores, tipografías, espacios, bordes, etc.) a documentos estructurados, como por ejemplo, páginas HTML o XML. El objetivo de la definición de este estándar es permitir la separación entre las normas de presentación y el propio contenido a mostrar.

HTML (HyperText Markup Language): se trata de un lenguaje basado en marcas o etiquetas, diseñado para la creación y publicación de páginas Web. HTML es una sintaxis para definir los elementos que se quieren mostrar en una página, y su estructura y ubicación, así como las relaciones entre ellos y con otros elementos de la Web mediante hipervínculos.

HTTP (HyperText Transfer Protocol): protocolo usado para acceder a la Web (WWW) que se encarga de procesar y dar respuestas a las peticiones para visualizar una página Web.

IEEE (The Institute of Electrical and Electronics Engineers): es una asociación de profesionales con sede en EEUU que fue fundada en 1884, que actualmente cuenta con miembros de más de 140 países. Investiga en campos como el aeroespacial, computacional, comunicaciones, etc. Y es muy conocida por ser un gran promotor de estándares.

JavaScript: es un lenguaje de scripting orientado a objetos, utilizado para acceder a objetos en aplicaciones. Principalmente, se utiliza integrado en un navegador web permitiendo el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas.

Login: momento de autenticación al ingresar a un servicio o sistema. En el momento que se inicia el login, el usuario entra en una sesión, empleando usualmente un nombre de usuario y contraseña.

MySQL: es un Sistema de Gestión de Bases de Datos multiusuario, multiplataforma y de código abierto que pertenece a la compañía sueca MySQLAB.

Password: palabra de acceso o contraseña que se utiliza junto con el login para identificarse en un sistema.

Página Web: fuente de información compatible con la WWW y que puede ser accedida a través de un navegador en Internet.

PHP (PHP Hypertext Pre-processor): es un lenguaje de programación interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. Este lenguaje es usado normalmente para la creación de páginas Web dinámicas.

Sitio Web: conjunto de páginas Web que están relacionadas entre sí, por lo general porque se accede desde un mismo dominio. Un sitio Web puede estar constituido por una o más páginas Web.

Proyecto: es el trabajo que el sistema controla y obtiene información de su desarrollo.

Usuario: en estDV hablaremos de Usuario como aquel que accede para usa el sistema (Alumno, Profesor o Administrador).

Profesor: así es como se designa a la persona que lleva el control de Alumnos y Proyectos, será un Usuario del sistema.

Alumno: es el rol destinado a los desarrolladores de los Proyectos almacenados en el sistema.

Administrador: rol asociado a la persona que está capacitada para controlar la aplicación. El Administrador pueden decide sobre los contenidos y funcionamiento de la aplicación, por ello deben conocer el uso de la aplicación con detalle.

Usuario Registrado: es el rol que se le asigna a aquel usuario que no siendo Administrador ha decidido registrarse en el sistema. Normalmente cuando se hable de Usuario Registrado se entenderá que el usuario se ha validado en el sistema mediante la introducción de login y password.

Web: por éste término se suele conocer a WWW (World Wide Web), creado por el Centro Europeo de Investigación Nuclear como un sistema de intercambio de información y que Internet ha estandarizado. Supone un medio cómodo y elegante, basado en multimedia e hipertexto, para publicar información en la red. Inicial y básicamente se compone del protocolo HTTP y del lenguaje HTML.

Procesador	Intel(R) Pentium(R) processor T4400 (@ 2.2GHz)
Fabricante	Acer
Tipo de Sistema	Sistema operativo de 64 bits.
Memoria	4,00 GB
Disco Duro	320 GB
Pantalla	15,6" (Proporción panorámica)
Tarjeta Gráfica	ATI Mobility Radeon HD 4570 Series 512MB

Tabla 1.1: *Equipo Hardware empleado para el proyecto.*

Sistema Operativo	Ubuntu 10.10
Servidor Web	Servidor HTTP Apache 2.2.16
Sistema de Gestión de Bases de Datos	MySQL (Versión 5)
Programación en lenguaje PHP	PHP 5
Editor de código fuente	Gedit, NetBeans IDE 6.9
Navegador web	Mozilla Firefox
Tratamiento de imágenes	GIMP 2 y Moo0 ImageThumbnailer 1.12
Programa de edición de texto	L ^A T _E X

Tabla 1.2: *Equipo Software empleado para el proyecto.*

1.4. Equipo de Desarrollo

El equipo hardware con el cual se ha llevado a cabo la creación de dicho proyecto, se muestra en la *Tabla 1.1*. Del mismo modo, el equipo software necesario para su creación se muestra en la *Tabla 1.2*. Por otro lado, mencionar que para montar el servidor web, el intérprete de PHP y el motor de bases de datos de MySQL, he empleado la aplicación **AppServ**.

1.5. Visión General

La presente memoria está ordenada de forma que se pueda resumir el trabajo realizado en cada una de las fases del proceso de desarrollo del proyecto software presentado.

El resto del documento se compone de: el Desarrollo del Calendario, donde se incluye una planificación temporal del desarrollo del proyecto, seguimiento del desarrollo del proyecto y porcentaje de esfuerzo dedicado a las diferentes fases del proceso de desarrollo; una Descripción General del Proyecto, donde se dará una visión general del proyecto, ampliando el contenido de la introducción de esta memoria; la Especificación de los Requisitos del proyecto, donde se detallan todas las funciones y casos de usos del proyecto; el proceso de análisis y diseño del producto software; la codificación del producto, donde se incluye un

breve resumen de los aspectos más relevantes de la implementación de la aplicación software; un Resumen, donde se comenta lo más destacado del trabajo realizado; las Conclusiones, donde se valora globalmente el trabajo realizado y se comentan las posibles ampliaciones que pueden darse; los manuales de usuario e instalación, y los anexos y la bibliografía.

De una forma más detallada, la estructura de la memoria es la siguiente:

- **Capítulo 2: Desarrollo del Calendario**

En esta sección se incluye una planificación temporal del desarrollo del proyecto, seguimiento del desarrollo del proyecto y porcentaje de esfuerzo dedicado a las diferentes fases del proceso de desarrollo.

- **Capítulo 3: Descripción General del Proyecto**

En esta sección se dará una visión general del proyecto, ampliando el contenido de la anteriormente introducido.

- **Capítulo 4: Especificación de Requisitos**

Este apartado contiene una lista detallada y completa de los requisitos que debe cumplir el software (funciones, prestaciones, restricciones de diseño y atributos) y de los requisitos de las interfaces del software con el resto de los elementos del sistema y con el exterior.

- **Capítulo 5: Análisis del Sistema**

En el análisis del sistema, se realizan modelos que ayudan a analizar y especificar la función, información y el comportamiento del sistema.

- **Capítulo 6: Diseño del Sistema**

Esta sección contempla el Diseño del Sistema de Información, que obtiene como resultado la estructura modular y los detalles procedimentales del sistema partiendo de lo obtenido en la fase de análisis.

- **Capítulo 7: Codificación**

En esta sección, se incluye un breve resumen de los aspectos más relevantes de la implementación de la aplicación software.

- **Capítulo 8: Resumen**

Esta sección incluye un breve resumen de lo más destacable del trabajo realizado.

- **Capítulo 9: Conclusiones**

En este capítulo se desarrollan las conclusiones a las que se ha llegado tras la elaboración del proyecto, las cuales contendrán una valoración global del trabajo realizado. Asimismo se contemplan las posibles ampliaciones que se pueden derivar del proyecto elaborado para la construcción de una posible futura y mejor aplicación.

- **Apéndice A: Manual de Usuario**

Contiene las explicaciones necesarias para dar asistencia al usuario y así ser capaz de manejar la aplicación.

- **Apéndice B: Manual de Instalación**

Contiene las explicaciones necesarias para dar asistencia al usuario y así ser capaz de instalar la aplicación.

- **Anexos**

- **Bibliografía**

- **Licencia: Licencia GNU Free Documentation License**

Capítulo 2

Desarrollo del Calendario

A través de las diferentes secciones de este apartado daremos una visión global de la planificación temporal del desarrollo del proyecto, del seguimiento de desarrollo del proyecto y del porcentaje de esfuerzo dedicado a las diferentes fases del proceso de desarrollo.

2.1. Actividades Mensuales

En esta sección hablaremos de las distintas actividades que se han llevado a cabo en el desarrollo del proyecto a lo largo de los distintos meses de trabajo. Mencionar que el proyecto se inició en Marzo de 2010 y ha finalizado en Julio de 2011, lo que supone aproximadamente un año y medio dedicado a la creación del proyecto, entre otras causas cabe destacar que esta prolongación ha sido debida a factores como: la finalización de las asignaturas de la titulación técnica, el cursar las asignaturas de cuarto curso (Ingeniería Informática) y el aprendizaje en programación web por el desconocimiento de los lenguajes de programación a usar (PHP, CSS,...). Cabe destacar que el proyecto se ha realizado con un modelo evolutivo, es decir, al principio no existía un enunciado claro del sistema por lo que los requisitos se han ido descubriendo a lo largo del desarrollo.

Dicho esto, mencionamos las distintas actividades desarrolladas mensualmente:

■ Marzo-Abril 2010:

- Reunión con el Cliente.
- Primeros pasos con CodeIgniter.

■ Mayo 2010:

- Primeros pasos en la gestión (Alta y Lista Usuario).

- Aprendizaje en CodeIgniter, PHP, HTML.
- Idea General del Proyecto.
- **Junio 2010:**
 - Parada (Exámenes).
- **Julio 2010:**
 - Se añade sesión de Usuario.
 - Añadido el Alumno a la BD.
 - Añadido el Proyecto a la BD.
 - Acceso de Alumnos al Sistema a través del DNI.
 - Añadidos Botones.
- **Agosto 2010:**
 - Relación entre Alumno-Proyecto y Alumno-Profesor.
 - Cambios en la relación entre Alumno-Proyecto.
 - BD terminada respecto a Alumnos, Profesores y Proyectos.
- **Septiembre-Octubre 2010:**
 - Exámenes.
 - Aprendizaje de SVN.
 - Aprendizaje de XML.
 - Aprendizaje de CSS.
- **Noviembre-Diciembre 2010, Enero 2011:**
 - Parada (Tiempo dedicado a Asignaturas y Exámenes)
- **Febrero 2011:**
 - Problemas de certificado con SVN.
 - Cambios en la relación entre Proyectos-Alumnos.
 - Aprender Tablas, formatos.
 - Decisión: No descargar proyecto (Mucho peso).
 - Aprender PHPlot.
 - Estadísticas como modelo.
- **Marzo 2011:**
 - Descarga de la información de forma Interactiva.

■ **Abril 2011:**

- Aplicación de PHPlot para las estadísticas.
- Comprobar la actualización de Proyectos.
- Establecer estadísticas que se van a mostrar.
- Parte de las estadísticas centradas en los Proyectos.
- Añadido a los proyectos una etiqueta de tipo (Web, juego...).

■ **Mayo 2011:**

- Aplicación de PHPlot para las estadísticas.
- Comprobar la actualización de Proyectos.
- Establecer estadísticas que se van a mostrar.
- Gráficas referentes a Proyectos.
- Gráficas referentes a Alumnos.
- Relación Alumno-Desarrollador.
- Añadida relación entre Proyecto-Profesor.
- Añadido todo lo referente a los Blog.

■ **Junio 2011:**

- Parada (Exámenes).

■ **Julio 2011:**

- Gráficas referentes a Alumnos y Proyectos.
- Añadida a la BD los datos de Administrador.
- Funciones de Administración terminadas.
- Seguridad de login.
- Posibilidad de descargar CSV de cada proyecto.
- Actualización automática a través de crontab.
- Arreglo de errores.
- Rediseñado de formularios.
- Seguridad en contraseñas (md5).
- Documentación.

■ **Agosto 2011:**

- Fin de la Documentación.

2.1.1. Diagrama de Gantt

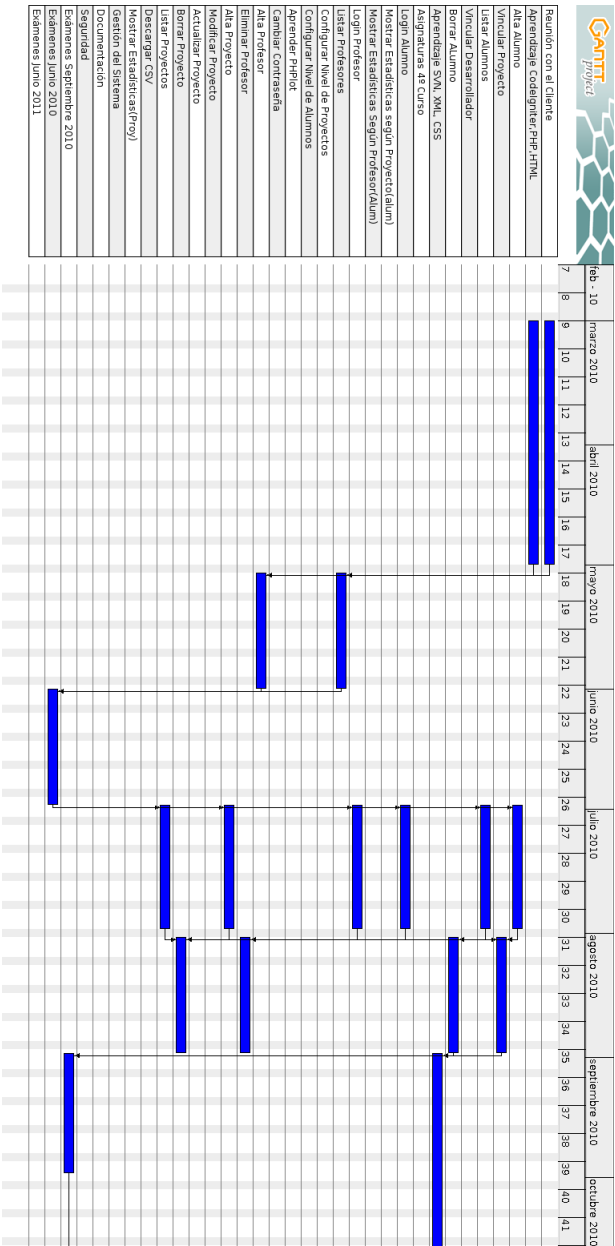


Figura 2.1: Diagrama de Gantt (1)

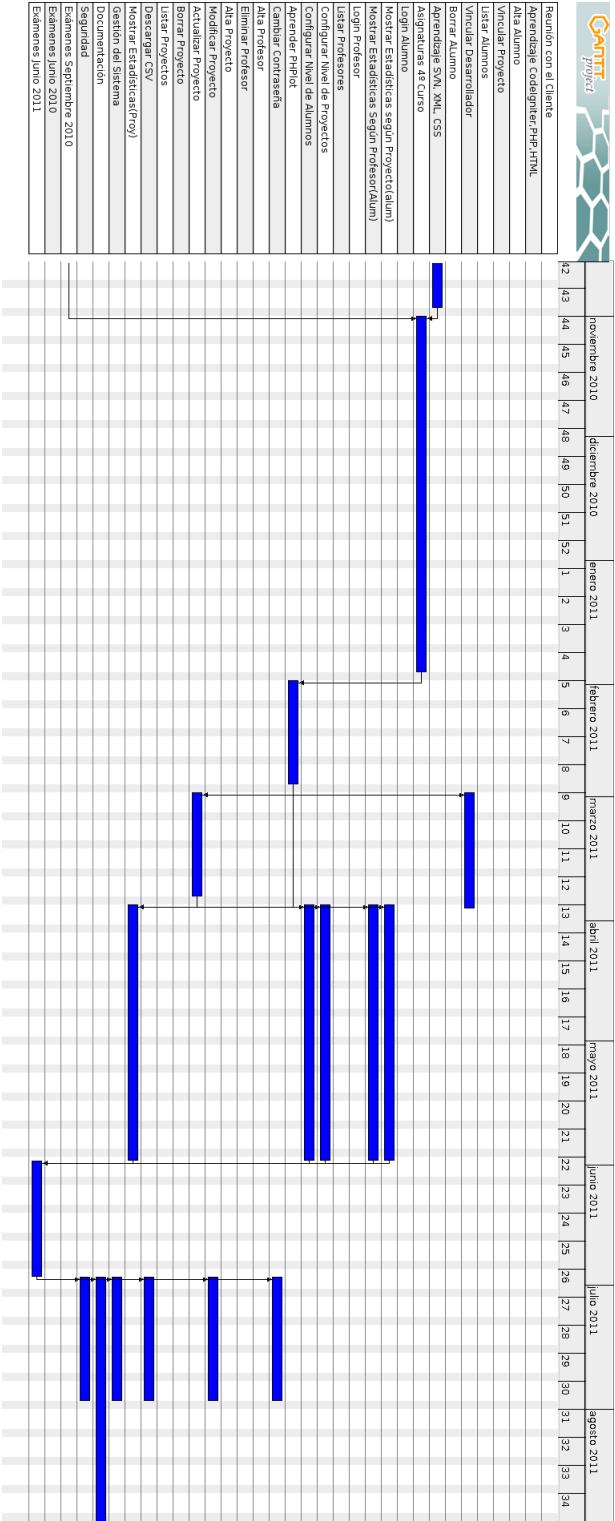


Figura 2.2: Diagrama de Gantt (2)

2.2. Crecimiento del Diagrama de Casos de Uso

En cuanto al crecimiento de la aplicación, en lo referente a funcionalidades, lo podemos observar en el gráfico de la *figura 2.1*, donde se muestran el número de casos de usos que se obtienen al final de cada mes, desde el comienzo del desarrollo hasta su final.

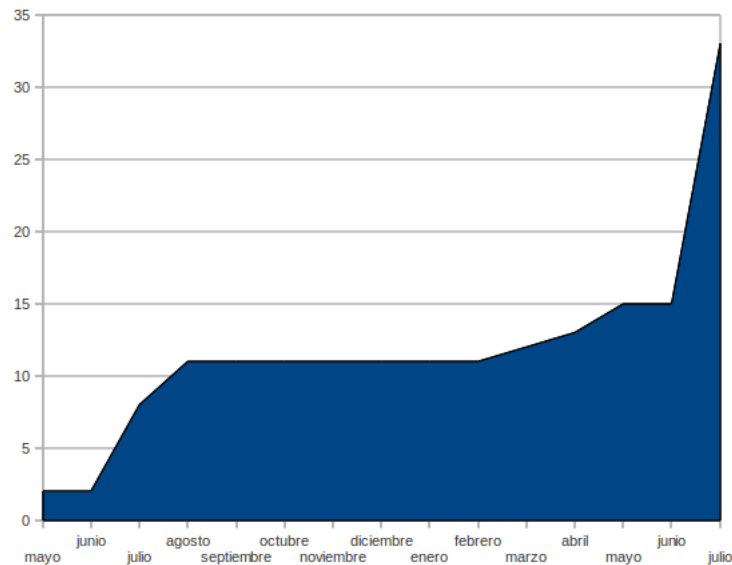


Figura 2.3: Crecimiento de los casos de uso.

Como se observa en el gráfico, partimos con una cantidad inicial de 2 casos de uso, que se corresponden con los primeros pasos en el la aplicación. Estos van aumentando a lo largo de los meses hasta finales de julio, donde nos encontramos con 33 casos de uso, que se corresponde con la aplicación completa y con todas sus funcionalidades. En la zona central del gráfico se ve como el número de casos de uso asciende poco, esto se debe a una mayor dedicación a la estructura de la aplicación que a sus funcionalidades.

2.3. Horas Dedicadas

Aquí vemos la relación de horas dedicadas al proyecto en cada mes. Se muestra de forma gráfica en la *figura 2.2*. Las horas que aquí se muestran son una estimación, ya que no he anotado detalladamente el número de horas y minutos dedicados de cada día. Pero son bastante fieles ya que mi dedicación al proyecto ha sido bastante regular y, además, las horas dedicadas se balancean por haber días en los que se dedica más tiempo y otros menos.

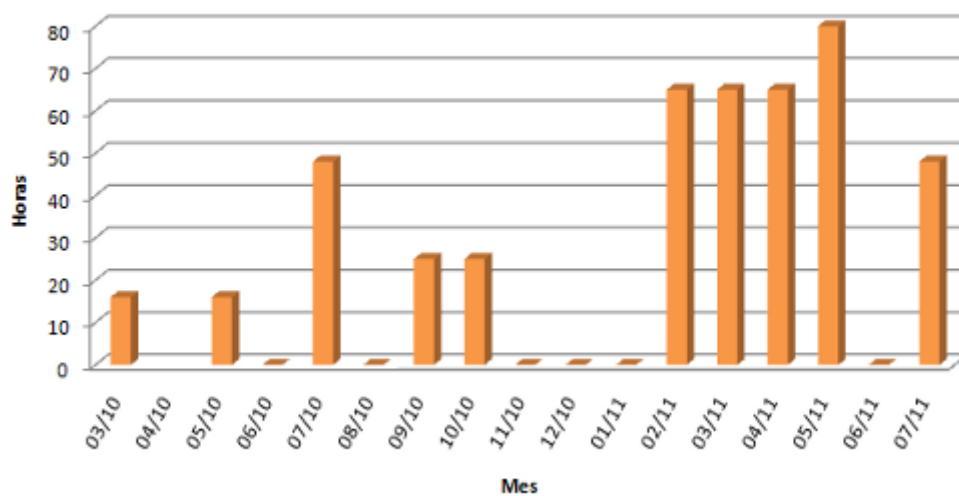


Figura 2.4: Horas Mensuales.

Podemos observar como el número de horas varía dependiendo del mes. Por ejemplo en los meses que coinciden con época de exámenes (Enero, Febrero, Junio), hubo una parada para dedicarlos al estudio de las diferentes asignaturas.

También podemos distinguir tres etapas importantes en el desarrollo del proyecto:

- **Aprendizaje Inicial:** Esta se extiende por todo el proyecto, pero está mayor arraigada a los primeros meses de desarrollo. En estos meses tuve que aprender mucho sobre programación Web. Esta etapa coincidió con mis últimas asignaturas de la titulación y el verano.
- **Desarrollo Intermedio:** Esta etapa comprende los meses correspondientes al primer cuatrimestre y el inicio del segundo, en los que debido a una carga excesiva de trabajo (asignaturas de Ingeniería Informática) el desarrollo del proyecto se vio bastante ralentizado.

- Desarrollo Final: Esta parte comprende la etapa final. En esta etapa la carga externa de trabajo fue menor, y eso se refleja en un aumento de horas dedicadas al PFC.

No se podría distinguir entre horas dedicadas a distintas partes del proyecto ya que el desarrollo ha sido bastante equilibrado en todas las partes del mismo. Las partes importantes podrían haber sido: aprendizaje, análisis, diseño, implementación y pruebas. Todas las partes se han ejecutado a la vez, de forma intercalada.

Capítulo 3

Descripción General

3.1. Perspectivas del Producto

Comentar que este proyecto **no es continuación** de ningún otro proyecto. Aunque se ha de tener en cuenta la existencia otro trabajo cuya existencia ha marcado la idea de este nuevo proyecto. Este trabajo se llama StatSVN y se encarga de obtener estadísticas de la información recogida en un repositorio SVN, el cual es el fundamento de estDV.

Como se especificó en apartados anteriores de este documento, el Sistema Operativo en el que se ha implantado el producto ha sido Linux (En concreto Ubuntu 10.10). EstDV está pensado y programado para funcionar sobre Linux, cualquier otro OS haría que no pudiera funcionar correctamente.

En cuanto a la **Interfaz del Usuario**, se ha pensado para ser lo más sencilla y accesible posible. La mejor solución: **Interfaz Web**, accesible simplemente con conexión a Internet y un navegador. Se ha tenido especial cuidado en la facilidad de uso y su comodidad para poder ser usado sin necesidad de amplios conocimientos informáticos. Este es uno de los objetivos básicos del proyecto.

Como **Interfaces Hardware**, nombrar aquéllas usuales en la utilización de cualquier programa, como pueden ser: el monitor, el teclado y ratón para la interacción habitual entre el usuario y el ordenador.

La página web, cuyo objetivo es acceder a la información obtenida a partir del repositorio SVN por parte de Alumnos y Profesores, la podemos dividir en tres grandes secciones, cada una orientada a un rol dentro del sistema (Alumnos, Profesores y Administrador).

El Alumno podrá acceder a: *Inicio*, donde se da la bienvenida al sitio web; *Proyectos*, donde podrá ver los proyectos en los que trabaja y hacer que actualicen o ver las gráficas asociadas; *Seguimiento*, donde podrá ver las gráficas asociadas a su trabajo personal.

El Profesor podrá acceder a: *Inicio*, donde se da la bienvenida al sitio web; *Alta Alumno*, donde podrá dar de alta un alumno para comenzar a recopilar estadísticas sobre él; *Alta Proyecto*, donde podrá dar de alta un proyecto para comenzar a recopilar estadísticas sobre él; *Mostrar Alumnos*, donde podrá ver los alumnos asociados a él y podrá ver sus gráficas asociadas según el alumno o según el alumno dentro de un proyecto. También podrá vincular un proyecto del sistema, vincular un desarrollador, ver sus proyectos asociados o borrarlo; *Mostrar Proyectos*, donde podrá ver los proyectos asociados e ir a la sección de gestión donde podrá actualizarlo, borrarlo, modificarlo o ver sus estadísticas; *Configuración*, donde podrá cambiar algunos datos de configuración para ajustar algunas opciones de estDV.

Por otro lado, el zona de administración, presenta una **Interfaz** dividida en 7 secciones: *Profesores*, *Alumnos*, *Proyectos*, *Nuevas Cuentas*, *Actualizar*, *Copia de Seguridad*, *Configuración*. Cada una de las cuales, posee diferentes opciones, relacionadas con la sección, las cuales permitirán llevar a cabo todas las funciones de administración (borrar, actualizar todo, configurar la actualización automática, dar de alta nuevas cuentas) en la aplicación.

Por último añadir que la parte de administración se trata de una herramienta a la que se accede por medio de un nombre de usuario y una contraseña desde la página inicial (sin ninguna otra sesión iniciada). Al igual que la zona dedicada a profesores.

El producto se encuentra desarrollado en HTML junto con el lenguaje de programación PHP, con el fin de poder acceder a la Base de Datos y generar contenido dinámico. Se accede a dicha Base de Datos MySQL para obtener los datos necesarios para generar dichas visualizaciones. Se ha necesitado por tanto trabajar con un sistema que ofrezca un Servidor Web con soporte para PHP junto con una Base de Datos MySQL. La mejor opción ha sido un Servidor Web Apache sobre la máquina descrita en apartados anteriores, permitiendo tener en la misma máquina tanto el Servidor Web como la Base de Datos MySQL; simulando la conexión entre un terminal y el Servidor Web a través de la Red.

Una característica importante de estDV es su comunicación con otros sistemas. Como se ha explicado anteriormente las estadísticas recopiladas por el sistema son obtenidas a través de un repositorio de Subversion, por lo que la comunicación con este repositorio es vital. Esta comunicación se ha establecido a través de la librería SVN de PHP.

3.2. Funciones del Producto

En la aplicación desarrollada, podemos distinguir tres grupos de acciones, una por cada rol del sistema. Por un lado, tenemos aquellas a las que pueden acceder los **alumnos** que visitan la página web:

1. Seguimiento Alumno.
2. Seguimiento Proyecto.
3. Seguimiento Alumno según Proyecto.
4. Actualizar Proyecto.

Por otro parte, están las que se puede acceder siendo **profesor**, con una cuenta de acceso (login y pass). Los profesores deberán registrarse en la aplicación y el administrador aceptará o denegará la cuenta. Las funciones a las que accede el profesor son:

- Crear, Listar, Ver Estadísticas, Actualizar o Borrar Proyectos.
- Crear, Listar, Ver Estadísticas o Borrar Alumnos.
- Asociar Proyectos y Alumnos.
- Configurar el sistema (cambiar contraseña, cambiar el número de revisiones por semana y mes)

Por último, tenemos las funciones a las que accederá el **administrador** del sistema. Estas funciones se basan en acciones cuyo objetivo es dar un buen funcionamiento al sistema:

- Listar o Borrar Proyectos.
- Listar o Borrar Alumnos.
- Listar o Borrar Profesores.
- Aceptar o Rechazar nuevas cuentas de Profesores.
- Actualizar todos los Proyectos.
- Crear Copias de Seguridad.
- Configurar el sistema (Momento de la actualización automática, cambiar contraseña).

3.3. Características del Usuario

Como hemos visto antes, existen tres tipos de usuarios: *Alumno*, *Profesor* y *Administrador*. Los requisitos para el acceso de cada uno de ellos es distinto en el sistema.

Por ejemplo el *Alumno* no necesita solicitar alta en el sistema, tampoco necesita tener una contraseña personal. Un alumno podrá acceder al sistema a través de su DNI siempre y cuando un *Profesor* le haya dado de alta en el sistema.

Por otro lado, el *Profesor* si ha de solicitar una cuenta personal, que el administrador aceptará o rechazará. Esta cuenta se identificará por un nombre de acceso (login) y una contraseña (password), con las cuales podrá acceder al sistema. Cada profesor solo podrá ver en el sistema los alumnos y proyectos que él haya creado.

Por último, el *Administrador*. Solo existe uno en el sistema (al menos una sola cuenta de administrador). Viene creada por defecto y podrá ser configurada en la zona de administración. Esta cuenta también se compone de nombre (login) y contraseña (password).

Hay que aclarar que los roles Alumno y Profesor, son solo roles. Es decir, cualquier persona puede darse de alta en el sistema como Profesor, siempre y cuando el Administrador lo permita, sin necesidad de ser profesor. De la misma forma cualquier Profesor podrá dar de alta en el sistema a otras personas como Alumnos sin necesidad de que sean alumnos. Estos roles se obtienen partiendo de la base de que estDV es una aplicación con objetivos académicos.

3.4. Restricciones Generales

Solo existe un requisito básico para poder acceder a la aplicación, es indispensable disponer de conexión a internet. Con conexión a internet y un navegador web podremos acceder al sistema y usarlo en su plenitud sin ningún problema.

Así pues, los requisitos mínimos necesarios para trabajar con estDV será una conexión a Internet y un navegador Web. En principio, el sistema es accesible desde cualquier navegador Web y por lo tanto, desde cualquier sistema operativo.

3.5. Suposiciones y Dependencias

La aplicación está diseñada para el acceso a través de un navegador Web compatible con el protocolo HTTP.

El host donde se ejecutará la aplicación estDV debe cumplir los siguientes requisitos:

- Permitir la ejecución de código en lenguaje PHP (Librería SVN y GD).
- Trabajar con bases de datos MySQL.

3.6. Requisitos para Futuras Versiones del Sistema

En futuras versiones sería interesante continuar con el formato utilizado en la interfaz del programa, no variando excesivamente la apariencia de las diferentes ventanas y la organización y posición de las diferentes funciones. Esto facilitará la rápida adaptación del usuario (siempre y cuando haya conocido y tratado versiones anteriores). Además de mantener todas las funcionalidades que ahora tiene el sistema.

Capítulo 4

Especificación de Requisitos

4.1. Introducción

En los siguientes apartados se definirá toda la documentación referente a los requisitos funcionales del software. *Se trata del resultado del trabajo conjunto entre los analistas/desarrolladores y los clientes/usuarios.* Con objetivo de tener una descripción precisa de qué desea el cliente.

La elaboración de estos requisitos es el resultado de recopilar la información obtenida tras las sesiones de trabajo realizadas con el cliente (por medio de entrevistas, prototipado, ...).

Aclarar que esta especificación de requisitos consiste en especificar **qué tiene que hacer el sistema**, incluida la interfaz, y no cómo desarrollarlo.

En concreto, para este proceso de especificación de requisitos se ha utilizado el estándar IEEE.

4.2. Metodología de Desarrollo

Antes de empezar con todo el proceso de desarrollo del software, vamos a comentar la metodología y el modelo de ciclo de vida empleados para llevar a cabo todo el trabajo.

Se ha empleado la metodología de planificación, desarrollo y mantenimiento de Sistemas de Información desarrollada por la Administración Pública española. Es decir, la *Metodología Métrica V.3*. Las causas de dicha elección, son entre otras que es la metodología vista en las asignaturas de la carrera, además de ser española y dar soporte a todas las etapas del ciclo de vida del software, o lo que es lo mismo, a todo el desarrollo en este documento descrito.

En cuanto al modelo de ciclo de vida de software o modelo de desarrollo empleado, he seguido el **Modelo Incremental**. Por ser un modelo evolutivo que nos permite ir adaptando el software a las necesidades que aparecen en el desarrollo, a través de pequeños incrementos. Además, uno de los factores principales en la decisión de escoger dicho modelo, fue la alta incertidumbre del comienzo, ya que los requisitos no estaban claramente definidos desde el comienzo del proyecto, sino que se han ido puliendo a lo largo del desarrollo.

A grandes rasgos, podemos decir, que el proyecto se inicio con un objetivo al que llegar. Esto hizo que los requisitos fueran apareciendo según nos acercábamos a dicho objetivo. Esto provocó que el sistema estuviera lleno de incrementos, podemos indicarlos a grandes rasgos:

- Incremento 1: Aparecen Alumnos, Proyectos y Usuarios.
- Incremento 2: Se cambia Usuario por Profesor y Usuario.
- Incremento 3: Se crean las funcionalidades para los Profesores.
- Incremento 4: Aparecen las estadísticas y las páginas de gráficas de proyecto.
- Incremento 5: Se añade una configuración de Profesor.
- Incremento 6: Se añade el Blog a las estadísticas.
- Incremento 7: Se añade una relación entre Alumno y Desarrollador.
- Incremento 8: Aparecen las gráficas de Alumnos y Alumnos-Proyectos.
- Incremento 9: Se crea la zona de administración.

4.3. Requisitos Específicos

4.3.1. Requisitos de Interfaces Externos

En este apartado se verán los requisitos específicos de la interfaces de: usuario, hardware, software y de comunicaciones.

Interfaces de Usuario

En el sistema, como hemos explicado anteriormente, existen tres zonas claramente distintas según el rol del usuario (Alumnos, Profesores y Administrador). Aun así, la estética del sitio es la misma para todos los casos, haciendo que sea equilibrada visualmente. Además, uno de los objetivos básicos del sistema es la claridad y comodidad de a la hora de usarla, debido a que esta es la gran mejora respecto a otras aplicaciones parecidas (StatSVN).

El diseño de las interfaces de los usuarios se centra en resultar agradable y de fácil manejo; de forma que las herramientas a disposición del usuario estarán en menús intuitivos, cómodos, fáciles de acceder y volver al inicio. Todos los menús se encontrarán en la parte superior de la ventana, dejando el resto para el contenido de la zona en la que estemos.

Por lo tanto, la aplicación va a tener un marco constante en el que solo cambiarán los lugares a los que podemos ir. Siguiendo siempre la misma estética hace que la aplicación sea, sobre todo, agradable a la vista, además de haber usado unos colores poco agresivos para que no sea molesto navegar por el sistema.

Interfaces Hardware

En esta sección vamos a diferenciar entre las necesidades hardware de **alojamiento de la aplicación** y las necesidades hardware de **acceso a la aplicación** por parte de los usuarios.

Para el **alojamiento de la aplicación** se recomienda un servicio de Hosting (alojamiento web) externo, que proporcione las interfaces software para el host recomendadas en el apartado Interfaces software.

El servicio de **alojamiento particular**, aunque no es aconsejable, es posible siempre que el hardware del servidor sea capaz de ejecutar los requisitos software mínimos especificados. Un motivo particular para evitar el alojamiento particular es el coste derivado de mantener una conexión adecuada para el servicio (conexión simétrica mínima de 1 Mbps). Normalmente estas conexiones sobrepasan el coste económico de contratar los servicios externos. Además, los servicios externos ofrecen ventajas como la de ocuparse de las caídas del servidor o de recuperar las pérdidas de datos.

Interfaces Software

Interfaz Software para el Hosting:

- **Plataformas:** Unix, GNU/Linux.
- **Requisitos de Alojamiento:** equipo con capacidad para ejecutar código PHP (Librerías SVN y GD).
- **Servidor de Bases de Datos:** MySQL.
- **Software Servidor:** Apache.

Interfaz Software para el Acceso:

- **Plataformas:** Unix, GNU/Linux, Windows, MAC OS.
- **Navegadores Recomendados:** Internet Explorer, Mozilla Firefox, Google Chrome.

Interfaces de Comunicaciones

La solución ofrecida por estDV se trata de una aplicación cliente-servidor que trabaja con el protocolo HTTP sobre conexiones TCP/IP.

4.3.2. Requisitos Funcionales

A continuación se presentan las descripciones de las diferentes funciones de las que dispone el sistema.

Gestión de Profesores

Las funciones que permitirán llevar a cabo el mantenimiento de los datos referentes a los mismos son:

- **Alta Profesor:** los datos relativos a un profesor: *Nombre, Apellidos, contraseña, email, login*; se registran en la Base de Datos.
- **Eliminar Profesor:** elimina un profesor y todo lo que haya creado (proyectos y alumnos) del sistema.

- **Listar Profesores:** muestra una lista con los profesores que existen en el sistema.
- **Cambiar Contraseña:** cambia la contraseña que usa para entrar al sistema.
- **Configurar Nivel Proyectos:** determina el número de revisiones y entradas al blog por mes y semana mínimos en un proyecto.
- **Configurar Nivel Alumnos:** determina el número de revisiones por mes y semana mínimos en un alumno.
- **Login Profesor:** inicia una sesión de Profesor en el sistema.

Gestión de Proyectos

Las funciones que permitirán llevar a cabo el mantenimiento de los datos referentes a los proyectos almacenados en el sistema son:

- **Alta Proyecto:** los datos relativos a un proyecto: *Nombre, Web, Blog, RSS del Blog, Dirección del repositorio SVN*; se registran en la Base de Datos.
- **Modificar Proyecto:** se permite cambiar algunos datos del proyecto: *Web, Blog, RSS del Blog, Dirección del repositorio SVN*.
- **Borrar Proyecto:** elimina un proyecto y las estadísticas guardadas sobre él.
- **Listar Proyecto:** muestra una lista con los proyectos que existen en el sistema relacionados con el Profesor que efectúa la operación.
- **Listar Todos los Proyectos:** muestra una lista con todos los proyectos que existen en el sistema.
- **Actualizar Proyecto:** conecta con el repositorio SVN y actualiza la base de datos con la información recopilada de cada revisión.
- **Mostar Estadísticas (Proy):** se muestran una serie de gráficas que describen la información recopilada en las actualizaciones centradas en el Proyecto.
- **Descargar CSV:** se descarga un archivo de extensión .csv con todos los datos almacenados en el sistema sobre el proyecto.

Gestión de Alumnos

Las funciones que nos permiten gestionar los datos referentes a los alumnos en el sistema son:

- **Alta Alumno:** los datos relativos a un alumno: *Nombre, Apellidos, DNI, Web, email*; se registran en la Base de Datos.
- **Borrar Alumno:** elimina un alumno del sistema.
- **Listar Alumnos:** una vez no interese seguir teniendo almacenado al cliente, el sistema permite eliminarlo.
- **Listar Todos los Alumnos:** muestra una lista con todos los alumnos que existen en el sistema.
- **Vincular Proyecto:** permite asociar el alumno con un proyecto creado por el mismo profesor.
- **Vincular Desarrollador:** se asocia el alumno con uno de los desarrolladores que se detectan en sus proyectos asociados.
- **Mostrar Estadísticas según Profesor:** se muestran una serie de gráficas que describen la información recopilada en las actualizaciones centradas en el Alumno y un Profesor.
- **Mostrar Estadísticas según Proyecto:** se muestran una serie de gráficas que describen la información recopilada en las actualizaciones centradas en el Alumno y un Proyecto.
- **Login Alumno:** inicia una sesión de Alumno en el sistema.

Gestión del Sistema

Con la gestión del sistema, introducimos funciones que permiten mantener un cierto control de los datos del sistema.

Podemos destacar la realización de copias de seguridad. Ya que es muy importante tener a salvo los datos por lo que pudiese ocurrir en el servidor que almacena la aplicación.

Por tanto, las funciones que contribuyen al mantenimiento del sistema por parte del Administrador son las siguientes:

- **Actualizar Todo:** esta función permitirá al administrador poner todos los proyectos en cola para que se actualicen.

- **Configurar Actualización:** aquí se podrá definir cada cuanto tiempo se efectuará una actualización de todos los proyectos de forma automática.
- **Listar Cuentas Nuevas:** esta función permitirá al administrador, eliminar los usuarios registrados de la página web, que desee.
- **Listar Copias de Seguridad:** esta función permitirá al administrador, eliminar los usuarios registrados de la página web, que desee.
- **Crear Copia de Seguridad:** esta función crea una copia de seguridad que se almacena en el sistema y se puede descargar.
- **Descargar Copia de Seguridad:** esta función permite descargar una copia de seguridad que se almacena en el sistema.
- **Borrar Copia de Seguridad:** esta función borra una copia de seguridad que se almacena en el sistema.
- **Login Administrador:** inicia una sesión de Administrador en el sistema.
- **Cambiar Contraseña Administrador:** cambia la contraseña de acceso al sistema del Administrador.

Identificación en el Sistema

Como se especificó en apartados anteriores de este documento, para poder hacer uso de la aplicación es requisito indispensable por parte del Administrador o Profesor poseer un nombre de usuario y una contraseña. Esto exige una autenticación al acceder a la aplicación. Del mismo modo un Alumno accederá al sistema a través de su DNI.

Así pues, existe una función de identificación que sirve como vía de acceso a la zona específica de cada tipo de usuario. Necesita, como es habitual: *nombre del usuario y contraseña* o *DNI* para los alumnos.

Cerrar Sesión

Del mismo modo que iniciamos una sesión, necesitamos tener una función que nos permita cerrarla, para controlar que otra persona tenga acceso al sistema. Dicha función, cierra la sesión iniciada por el usuario al identificarse y así asegura el acceso no autorizado. Esta función es la misma para Alumnos, Profesores y Administrador.

4.3.3. Requisitos de Rendimiento

No han sido considerados requisitos de rendimiento como algo fundamental en nuestro proyecto.

4.3.4. Otros Requisitos

La aplicación trabajará sobre una base de datos MySQL donde se almacenará toda la información necesaria para el funcionamiento de la misma. El sistema contiene funciones con las modificar la base de datos (añadir, eliminar, ...) de forma segura. Esto elimina riesgos y hace que cualquier persona sin formación respecto a base de datos pueda utilizarlo.

Capítulo 5

Análisis del Sistema

5.1. Introducción

En la Fase de Análisis y Diseño del Proyecto se estudia toda la funcionalidad de la aplicación a desarrollar. A lo largo de esta fase se hace uso de diferentes modelos y diagramas que ayudan a analizar y especificar la función, información y el comportamiento del sistema.

5.2. Diagramas de Casos de Uso

En este apartado vamos a describir el análisis completo de los casos de uso obtenidos del sistema. Los casos de usos nos ayudan a obtener los requisitos del sistema. Cada caso de uso muestra uno o más escenarios donde se indica como debe el sistema permitir que los actores interactuen con él (actor: entidad externa al sistema que guarda relación con éste y que demanda una funcionalidad) para conseguir el objetivo específico. Lo podríamos definir como una secuencia de acciones que se desarrollan entre el sistema y los actores en respuesta a un evento iniciado por un actor principal sobre el sistema. De este modo, mediante los diagramas de casos de uso se especifican la comunicación y el comportamiento del sistema en la interacción con los usuarios y/u otros sistemas.

5.2.1. Diagrama de Casos de Uso: Gestión de Alumnos

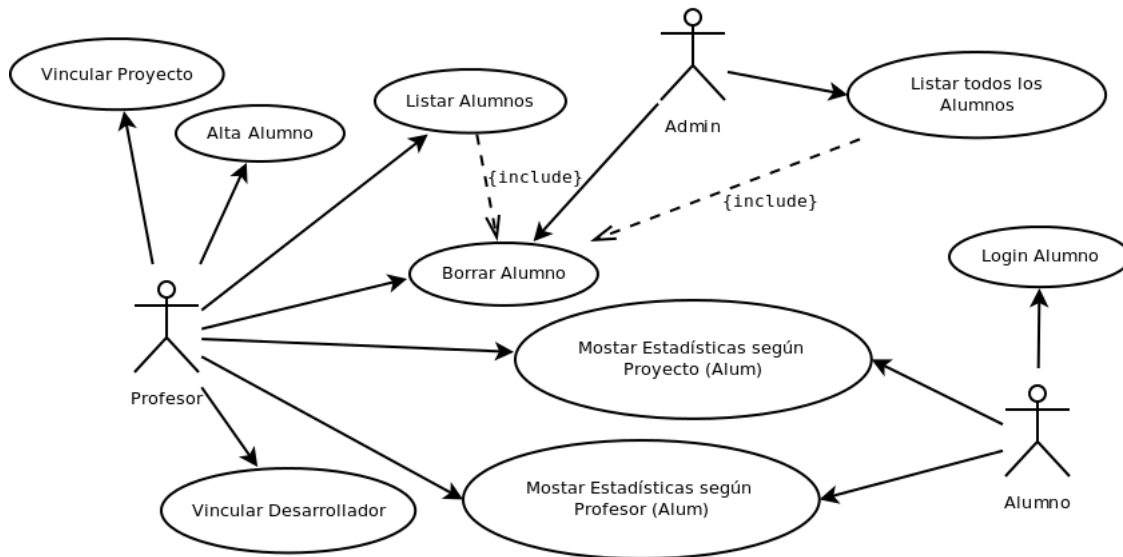


Figura 5.1: Diagrama de Casos de Uso: Gestión de Alumnos.

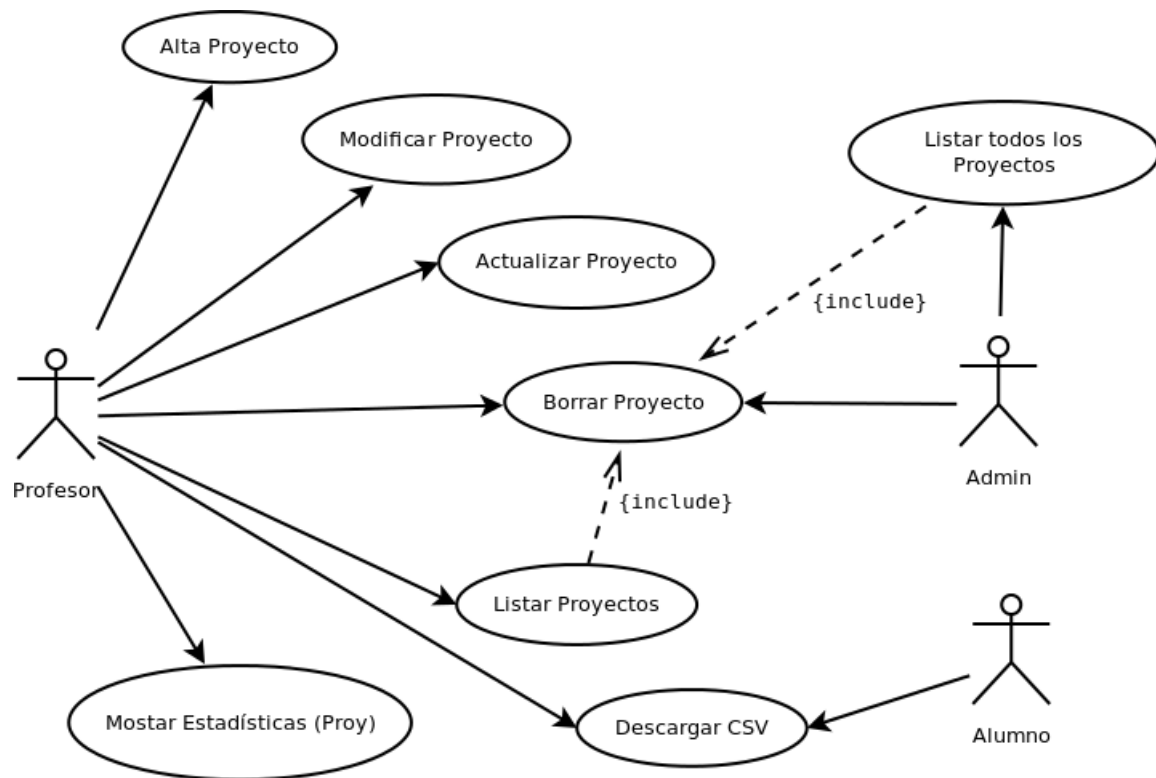
5.2.2. Diagrama de Casos de Uso: Gestión de Proyectos

Figura 5.2: Diagrama de Casos de Uso: Gestión de Proyectos.

5.2.3. Diagrama de Casos de Uso: Gestión de Profesores

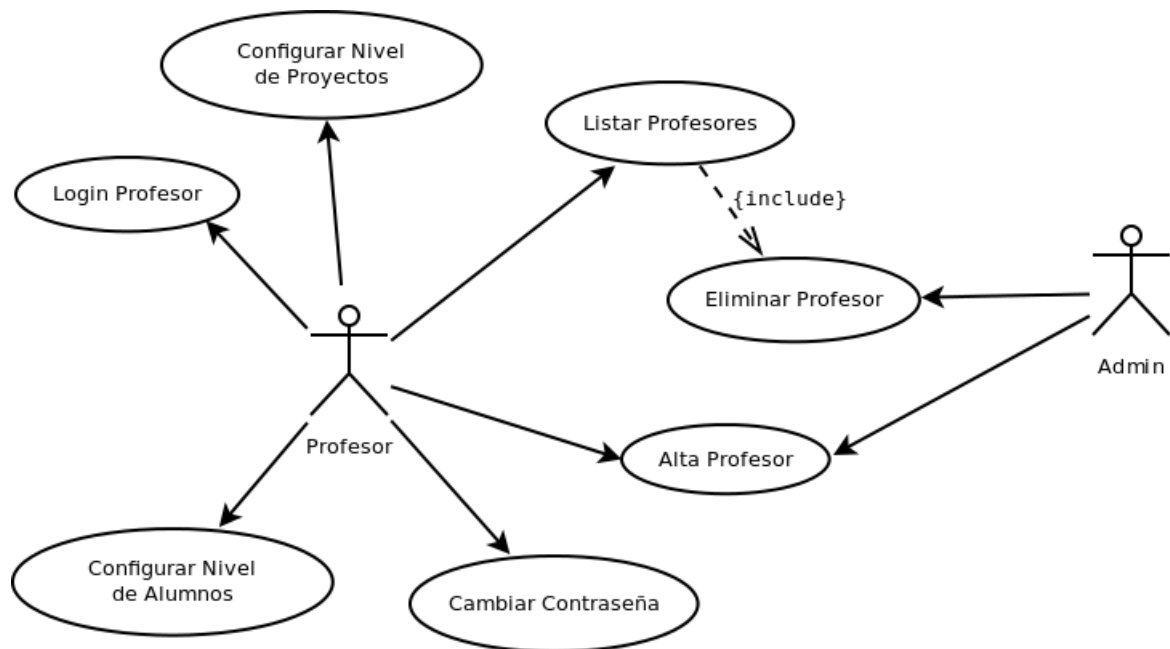


Figura 5.3: Diagrama de Casos de Uso: Gestión de Profesores.

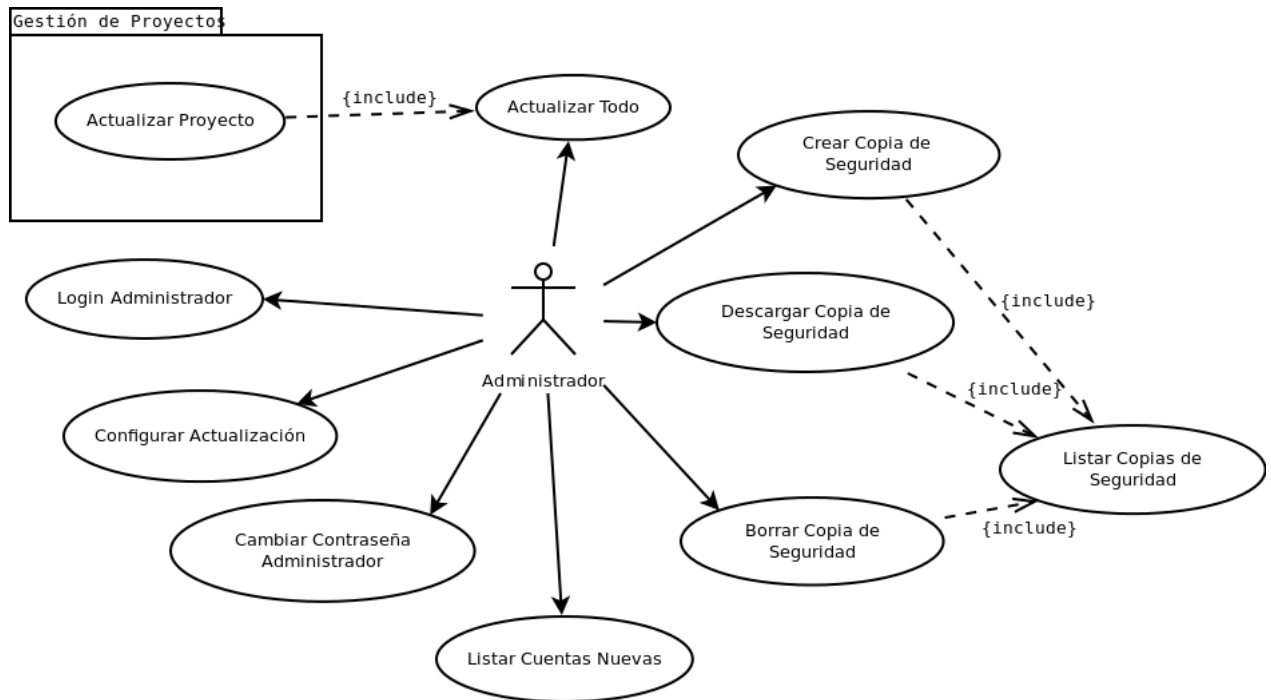
5.2.4. Diagrama de Casos de Uso: Gestión del Sistema

Figura 5.4: Diagrama de Casos de Uso: Gestión del Sistema.

5.3. Especificación de Casos de Uso

5.3.1. Especificación del Caso de Uso: Alta Alumno

Caso de Uso: ALTA ALUMNO

Actores Principales: Profesor

Actores Secundarios: —

Precondiciones: El Profesor está logueado en el sistema.

Postcondiciones: El Alumno queda registrado en el sistema.

Escenario principal:

1. El Profesor introduce el Nombre, Apellidos, email, DNI y una web.
2. El sistema comprueba que el alumno no está registrado o asociado a dicho Profesor y lo registra.

Flujos alternativos:

- *a. En cualquier momento el Profesor cancela el registro:
 1. El sistema cancela la operación.
- 1a. El DNI indicado no es válido:
 1. El sistema lo notifica y reinicia la operación.
- 1b. El Alumno ya está registrado pero no está asociado al Profesor:
 1. El sistema lo asocia al Profesor.
- 1c. La dirección de correo introducida no es válida:
 1. El sistema lo notifica y reinicia la operación.
- 1d. El Alumno ya está registrado:
 1. El sistema lo notifica y reinicia la operación.
- 1e. El Profesor no introduce una web:
 1. El sistema registra al Alumno sin web personal.

5.3.2. Especificación del Caso de Uso: Listar Alumnos

Caso de Uso: LISTAR ALUMNOS

Actores Principales: Profesor

Actores Secundarios: —

Precondiciones: El Profesor está logueado.

Postcondiciones: Se muestra una lista con los Alumnos asociados al Profesor.

Escenario principal:

1. El Profesor pide al sistema la lista de los Alumnos que tiene asociado.
2. El sistema proporciona la lista de dichos Alumnos.

Flujos alternativos:

- 1a. El Profesor no tiene Alumnos asociados:
 1. El sistema proporcionará una lista vacía.

5.3.3. Especificación del Caso de Uso: Listar todos los Alumnos

Caso de Uso: LISTAR TODOS LOS ALUMNOS

Actores Principales: Administrador

Actores Secundarios: —

Precondiciones: El Administrador está logueado en el sistema.

Postcondiciones: Se muestra una lista con todos los Alumnos del Sistema.

Escenario principal:

1. El Administrador pide al sistema la lista de todos los Alumnos que existen.
2. El sistema proporciona la lista de dichos Alumnos.

Flujos alternativos:

- 1a. En el sistema no existen Alumnos registrados:
 1. El sistema proporcionará una lista vacía.

5.3.4. Especificación del Caso de Uso: Borrar Alumno

Caso de Uso: BORRAR ALUMNO

Actores Principales: Profesor, Administrador

Actores Secundarios: —

Precondiciones: El Profesor o el Administrador están logueados en el sistema.

Postcondiciones: El Alumno quedará eliminado del sistema.

Escenario principal:

1. Include: Listar Alumnos
2. El Profesor selecciona al Alumno que desea eliminar.
3. El sistema pide confirmación de que desea eliminar a dicho Alumno.
4. El Profesor afirma que desea eliminarlo.
5. El sistema elimina al Alumno del sistema.

Flujos alternativos:

- *a. En cualquier momento el Profesor cancela la eliminación de Alumnos.
 1. El sistema cancela la operación.
- 1a. Si el que ejecuta el caso de uso es el Administrador:
 1. Include: Listar todos los Alumnos
 2. El Administrador selecciona al Alumno que desea eliminar.
 3. El sistema pide confirmación de que desea eliminar a dicho Alumno.
 4. El Administrador afirma que desea eliminarlo.
 5. El sistema elimina al Alumno del sistema.

5.3.5. Especificación del Caso de Uso: Vincular Proyecto

Caso de Uso: VINCULAR PROYECTO

Actores Principales: Profesor

Actores Secundarios: —

Precondiciones: El Profesor está logueado en el sistema.

Postcondiciones: Se asigna el Alumno como desarrollador del Proyecto.

Escenario principal:

1. Include: Listar Alumnos
2. El Profesor selecciona el Alumno al que desea asociar con un Proyecto.
3. Include: Listar Proyectos
4. El Profesor selecciona el Proyecto con el que desea asociar al Alumno.
5. El sistema registra al Alumno como desarrollador en dicho Proyecto.

Flujos alternativos:

- *a. En cualquier momento el Profesor cancela la operación:
 1. El sistema cancela la operación.

5.3.6. Especificación del Caso de Uso: Vincular Desarrollador

Caso de Uso: VINCULAR DESARROLLADOR

Actores Principales: Profesor

Actores Secundarios: —

Precondiciones: El Profesor está logueado en el sistema.

Postcondiciones: Se asocia al Alumno como un desarrollador concreto de un Proyecto.

Escenario principal:

1. Include: Listar Alumnos
2. El Profesor selecciona el Alumno al que desea asociar con un Desarrollador.
3. El sistema muestra una lista de desarrolladores (Nombres) de sus proyectos.
4. El Profesor selecciona el desarrollador con el que desea asociar al Alumno.
5. El sistema registra al Alumno como ese desarrollador en dicho Proyecto.

Flujos alternativos:

- *a. En cualquier momento el Profesor cancela la operación:
 1. El sistema cancela la operación.
- 2a. El Alumno no tiene desarrolladores posibles:
 1. El sistema muestra una lista vacía.

5.3.7. Especificación del Caso de Uso: Mostrar Estadísticas según Profesor

Caso de Uso: MOSTRAR ESTADÍSTICAS SEGÚN PROFESOR

Actores Principales: Profesor, Alumno

Actores Secundarios: —

Precondiciones: El Profesor o el Alumno están logueados.

Postcondiciones: Se muestra una página con estadísticas recopiladas por el sistema.

Escenario principal:

1. Include: Listar Alumnos
2. El Profesor selecciona el Alumno para ver sus estadísticas.
3. El sistema muestra las estadísticas referentes al Alumno y al Profesor.

Flujos alternativo:

- *a. En cualquier momento el usuario cancela la operación:
 1. El sistema cancela la operación.
- 1a. El Actor es un Alumno:
 1. El Alumno selecciona las estadísticas referentes a si mismo y un Profesor.
 2. El sistema muestra una lista de Profesores asociados al Alumno.
 3. El Alumno selecciona uno de ellos.
 4. El sistema muestra las estadísticas referentes al Alumno y al Profesor.

5.3.8. Especificación del Caso de Uso: Mostrar Estadísticas según Proyecto

Caso de Uso: MOSTRAR ESTADÍSTICAS SEGÚN PROYECTO

Actores Principales: Profesor, Alumno

Actores Secundarios: —

Precondiciones: El Profesor o el Alumno están logueados.

Postcondiciones: Se muestra una página con estadísticas recopiladas por el sistema.

Escenario principal:

1. Include: Listar Alumnos
2. El Profesor selecciona el Alumno para ver sus estadísticas.
3. El sistema muestra los Proyectos asociados al Alumno.
4. EL Profesor selecciona uno de los proyectos.
5. El sistema muestra las estadísticas referentes al Alumno y al Proyecto.

Flujos alternativo:

- *a. En cualquier momento el usuario cancela la operación:
 1. El sistema cancela la operación.
- 1a. El Actor es un Alumno:
 1. El Alumno selecciona las estadísticas referentes a si mismo y un Proyecto.
 2. El sistema muestra una lista de Proyectos asociados al alumno.
 3. El Alumno selecciona uno de ellos.
 4. El sistema muestra las estadísticas referentes al Alumno y al Proyecto.

5.3.9. Especificación del Caso de Uso: Login Alumno

Caso de Uso: LOGIN ALUMNO

Actores Principales: Alumno

Actores Secundarios: —

Precondiciones: El Alumno no está logueado.

Postcondiciones: Se inicia una sesión de Alumno en el sistema.

Escenario principal:

1. El Alumno solicita iniciar sesión.
2. El sistema pide al Alumno que introduzca el DNI.
3. El Alumno introduce su DNI.
4. EL sistema comprueba que el DNI es correcto y que está registrado en el sistema e inicia la sesión de Alumno.

Flujos alternativo:

- *a. En cualquier momento el usuario cancela la operación:
 1. El sistema cancela la operación.
- 4a. El DNI no es correcto:
 1. El sistema lo notifica y reinicia la operación.
- 4b. El Alumno no está registrado en el sistema:
 1. El sistema lo notifica y reinicia la operación.

5.3.10. Especificación del Caso de Uso: Alta Proyecto

Caso de Uso: ALTA PROYECTO

Actores Principales: Profesor

Actores Secundarios: —

Precondiciones: El Profesor está logueado en el sistema.

Postcondiciones: El Proyecto queda registrado en el sistema.

Escenario principal:

1. El Profesor introduce el DNI del coordinador, Nombre, web, Blog, RSS del Blog y dirección del repositorio SVN.
2. El sistema comprueba que el Proyecto no está registrado y que el DNI es de un alumno asociado al Profesor y lo registra.

Flujos alternativos:

- *a. En cualquier momento el Profesor cancela el registro:
 1. El sistema cancela la operación.
- 1a. El DNI indicado no es válido:
 1. El sistema lo notifica y reinicia la operación.
- 1b. El Proyecto ya está registrado:
 1. El sistema lo notifica y reinicia la operación.
- 1c. El Profesor no introduce una web:
 1. El sistema registra el Proyecto sin web personal.

5.3.11. Especificación del Caso de Uso: Listar Proyectos

Caso de Uso: LISTAR PROYECTOS

Actores Principales: Profesor

Actores Secundarios: —

Precondiciones: El Profesor está logueado.

Postcondiciones: Se muestra una lista con los Proyectos asociados al Profesor.

Escenario principal:

1. El Profesor pide al sistema la lista de los Proyectos que tiene asociado.
2. El sistema proporciona la lista de dichos Proyectos.

Flujos alternativos:

- 1a. El Profesor no tiene Proyectos asociados:
 1. El sistema proporcionará una lista vacía.

5.3.12. Especificación del Caso de Uso: Listar todos los Proyectos

Caso de Uso: LISTAR TODOS LOS PROYECTOS

Actores Principales: Administrador

Actores Secundarios: —

Precondiciones: El Administrador está logueado en el sistema.

Postcondiciones: Se muestra una lista con todos los Proyectos del Sistema.

Escenario principal:

1. El Administrador pide al sistema la lista de todos los Proyectos que existen.
2. El sistema proporciona la lista de dichos Proyectos.

Flujos alternativos:

- 1a. En el sistema no existen Proyectos registrados:
 1. El sistema proporcionará una lista vacía.

5.3.13. Especificación del Caso de Uso: Borrar Proyecto

Caso de Uso: BORRAR PROYECTO

Actores Principales: Profesor, Administrador

Actores Secundarios: —

Precondiciones: El Profesor o el Administrador están logueados en el sistema.

Postcondiciones: El Proyecto quedará eliminado del sistema.

Escenario principal:

1. Include: Listar Proyectos
2. El Profesor selecciona el Proyecto que desea eliminar.
3. El sistema pide confirmación de que desea eliminar a dicho Proyecto.
4. El Profesor afirma que desea eliminarlo.
5. El sistema elimina el Proyecto del sistema.

Flujos alternativos:

- *a. En cualquier momento el Profesor cancela la eliminación de Proyectos.
 1. El sistema cancela la operación.
- 1a. Si el que ejecuta el caso de uso es el Administrador:
 1. Include: Listar todos los Proyectos
 2. El Administrador selecciona el Proyecto que desea eliminar.
 3. El sistema pide confirmación de que desea eliminar a dicho Proyecto.
 4. El Administrador afirma que desea eliminarlo.
 5. El sistema elimina el Proyecto del sistema.

5.3.14. Especificación del Caso de Uso: Mostrar Estadísticas (Proy)

Caso de Uso: MOSTRAR ESTADÍSTICAS (PROY)

Actores Principales: Profesor, Alumno

Actores Secundarios: —

Precondiciones: El Profesor o el Alumno están logueados.

Postcondiciones: Se muestra una página con estadísticas de un proyecto recopiladas por el sistema.

Escenario principal:

1. Include: Listar Proyectos
2. El Profesor selecciona el Proyecto para ver sus estadísticas.
3. El sistema muestra las estadísticas referentes al Proyecto.

Flujos alternativo:

- *a. En cualquier momento el usuario cancela la operación:
 1. El sistema cancela la operación.
- 1a. El Actor es un Alumno:
 1. El Alumno selecciona las estadísticas a un Proyecto.
 2. El sistema muestra una lista de Proyectos asociados al alumno.
 3. El Alumno selecciona uno de ellos.
 4. El sistema muestra las estadísticas referentes al Proyecto.

5.3.15. Especificación del Caso de Uso: Actualizar Proyecto

Caso de Uso: ACTUALIZAR PROYECTO

Actores Principales: Profesor, Alumno

Actores Secundarios: —

Precondiciones: El Profesor o el Alumno están logueados.

Postcondiciones: Se descargan los nuevos datos que existan en el repositorio SVN y el Blog.

Escenario principal:

1. Include: Listar Proyectos
2. El Profesor selecciona el Proyecto para empezar la descarga.
3. El sistema muestra la revisión actual del sistema y la del repositorio.
4. El Profesor selecciona la opción de descargar.
5. El sistema muestra una monitorización de la actualización del Proyecto.

Flujos alternativo:

- *a. En cualquier momento el usuario cancela la operación:
 1. El sistema cancela la operación.
- 1a. El Actor es un Alumno:
 1. El Alumno selecciona ver sus Proyectos.
 2. El sistema muestra una lista de Proyectos asociados al alumno.
 3. El Alumno selecciona uno de ellos.
 4. El sistema muestra la revisión actual del sistema y la del repositorio.
 5. El Alumno selecciona la opción de descargar.
 6. El sistema muestra una monitorización de la actualización del Proyecto.
- 3a. El repositorio no está disponible:
 1. El sistema mostrará el error y volverá al inicio.
- 5a. El Proyecto no tiene Blog:
 1. El sistema solo actualizará el repositorio Subversion.

5.3.16. Especificación del Caso de Uso: Descargar CSV

Caso de Uso: DESCARGAR CSV

Actores Principales: Profesor, Alumno

Actores Secundarios: —

Precondiciones: El Profesor o el Alumno están logueados.

Postcondiciones: Se descargan un archivo de extensión .csv con todos los datos almacenados en el sistema sobre el proyecto.

Escenario principal:

1. Include: Listar Proyectos
2. El Profesor selecciona el Proyecto para empezar la descarga.
3. El sistema muestra algo de información sobre el Proyecto.
4. El Profesor selecciona la opción de descargar CSV.
5. El sistema ofrece al Profesor la descarga del archivo.
6. El Profesor acepta la descarga.

Flujos alternativo:

- *a. En cualquier momento el usuario cancela la operación:
 1. El sistema cancela la operación.
- 1a. El Actor es un Alumno:
 1. El Alumno selecciona ver sus Proyectos.
 2. El sistema muestra una lista de Proyectos asociados al alumno.
 3. El Alumno selecciona uno de ellos.
 4. El sistema muestra algo de información sobre el Proyecto.
 5. El Alumno selecciona la opción de descargar CSV.
 6. El sistema ofrece al Alumno la descarga del archivo.
 7. El Alumno acepta la descarga.
- 3a. El Proyecto no tiene datos almacenados en el sistema:
 1. El sistema no permitirá la descarga del archivo CSV.

5.3.17. Especificación del Caso de Uso: Alta Profesor

Caso de Uso: ALTA PROFESOR

Actores Principales: Profesor

Actores Secundarios: Administrador

Precondiciones: El Profesor no está logueado ni registrado en el sistema.

Postcondiciones: El Profesor queda registrado en el sistema.

Escenario principal:

1. El Profesor introduce el Nombre, Apellidos, contraseña, email, login.
2. El sistema comprueba que el Profesor no está registrado y que el login es único, y lo marca como nueva cuenta.
3. El Administrador acepta la nueva cuenta.
4. El sistema registra la cuenta del Profesor como activa en el sistema.

Flujos alternativos:

- *a. En cualquier momento el Profesor cancela el registro:
 1. El sistema cancela la operación.
- 2a. El login indicado ya está siendo usado:
 1. El sistema lo notifica y reinicia la operación.
- 2b. El email no es correcto:
 1. El sistema lo notifica y reinicia la operación.
- 3a. El Administrador no acepta la cuenta:
 1. El sistema elimina la cuenta.

5.3.18. Especificación del Caso de Uso: Listar Profesores

Caso de Uso: LISTAR PROFESORES

Actores Principales: Administrador

Actores Secundarios: —

Precondiciones: El Administrador está logueado.

Postcondiciones: Se muestra una lista con los Profesores que existen en el sistema.

Escenario principal:

1. El Administrador pide al sistema la lista de los Profesores que tiene asociado.
2. El sistema proporciona la lista de dichos Profesores.

Flujos alternativos:

- 2a. No existen Profesores en el sistema:
 1. El sistema proporcionará una lista vacía.

5.3.19. Especificación del Caso de Uso: Borrar Profesor

Caso de Uso: BORRAR PROFESOR

Actores Principales: Administrador

Actores Secundarios: —

Precondiciones: El Administrador está logueado en el sistema.

Postcondiciones: El Proyecto quedá eliminado del sistema.

Escenario principal:

1. Include: Listar Profesores
2. El Administrador selecciona el Profesor que desea eliminar.
3. El sistema pide confirmación de que desea eliminar a dicho Profesor.
4. El Administrador afirma que desea eliminarlo.
5. El sistema elimina el Profesor del sistema.

Flujos alternativos:

- *a. En cualquier momento el Administrador cancela la eliminación de Profesores.
 1. El sistema cancela la operación.

5.3.20. Especificación del Caso de Uso: Cambiar Contraseña

Caso de Uso: CAMBIAR CONTRASEÑA

Actores Principales: Profesor

Actores Secundarios: —

Precondiciones: El Profesor está logueado en el sistema.

Postcondiciones: Asigna una nueva contraseña a la cuenta del Profesor.

Escenario principal:

1. El Profesor solicita cambiar al contraseña
2. El sistema pide la contraseña actual y la nueva.
3. El Profesor indica las contraseñas.
4. El sistema comprueba que las contraseñas sean correctas y la actualiza.

Flujos alternativos:

- *a. En cualquier momento el Profesor cancela la operación.
 1. El sistema cancela la operación.
- 4a. La contraseña actual no es correcta.
 1. El sistema lo notifica y reinicia la operación.
- 4b. La nueva contraseña no es correcta.
 1. El sistema lo notifica y reinicia la operación.

5.3.21. Especificación del Caso de Uso: Configurar Nivel Proyectos

Caso de Uso: CONFIGURAR NIVEL PROYECTOS

Actores Principales: Profesor

Actores Secundarios: —

Precondiciones: El Profesor está logueado en el sistema.

Postcondiciones: Asigna nuevos valores de nivel a la configuración del Profesor.

Escenario principal:

1. El Profesor solicita modificar los valores de nivel de Proyectos.
2. El sistema pide los nuevos datos (Revisiones por semana y mes y Entradas al Blog por semana y mes).
3. El Profesor indica los valores.
4. El sistema comprueba que sean correctos y lo actualiza.

Flujos alternativos:

- *a. En cualquier momento el Profesor cancela la operación.
 1. El sistema cancela la operación.
- 4a. No son valores correctos.
 1. El sistema lo notifica y reinicia la operación.

5.3.22. Especificación del Caso de Uso: Configurar Nivel Alumnos

Caso de Uso: CONFIGURAR NIVEL ALUMNOS

Actores Principales: Profesor

Actores Secundarios: —

Precondiciones: El Profesor está logueado en el sistema.

Postcondiciones: Asigna nuevos valores de nivel a la configuración del Profesor.

Escenario principal:

1. El Profesor solicita modificar los valores de nivel de Alumnos.
2. El sistema pide los nuevos datos (Revisiones por semana y mes).
3. El Profesor indica los valores.
4. El sistema comprueba que sean correctos y lo actualiza.

Flujos alternativos:

- *a. En cualquier momento el Profesor cancela la operación.
 1. El sistema cancela la operación.
- 4a. No son valores correctos.
 1. El sistema lo notifica y reinicia la operación.

5.3.23. Especificación del Caso de Uso: Login Profesor

Caso de Uso: LOGIN PROFESOR

Actores Principales: Profesor

Actores Secundarios: —

Precondiciones: El Profesor no está logueado.

Postcondiciones: Se inicia una sesión de Profesor en el sistema.

Escenario principal:

1. El Profesor solicita iniciar sesión.
2. El sistema pide al Profesor que introduzca el Login y Pass.
3. El Profesor introduce los datos.
4. EL sistema comprueba que son correctos y que está registrado en el sistema.

Flujos alternativo:

- *a. En cualquier momento el usuario cancela la operación:
 1. El sistema cancela la operación.
- 4a. El Login o Pass no son correctos:
 1. El sistema lo notifica y reinicia la operación.
- 4b. El Profesor no está registrado o no tiene su cuenta activada:
 1. El sistema lo notifica y reinicia la operación.

5.3.24. Especificación del Caso de Uso: Actualizar Todo

Caso de Uso: ACTUALIZAR TODO

Actores Principales: Administrador

Actores Secundarios: —

Precondiciones: El Administrador está logueado.

Postcondiciones: Actualiza los datos de todos los Proyectos del sistema.

Escenario principal:

1. El Administrador solicita ver los proyectos para actualizar.
2. El sistema muestra una lista de proyectos y la revisión actual en el sistema.
3. El Adminsitrador solicita la actualización.
4. Include: Actualizar Proyecto.
 - El sistema repite el último paso para cada Proyecto.

Flujos alternativo:

- *a. En cualquier momento el Administrador cancela la operación:
 1. El sistema cancela la operación.
- 2a. No existen Proyectos en el sistema:
 1. El sistema lo notifica y reinicia la operación.

5.3.25. Especificación del Caso de Uso: Configurar Actualización

Caso de Uso: CONFIGURAR ACTUALIZACIÓN

Actores Principales: Administrador

Actores Secundarios: —

Precondiciones: El Administrador está logueado en el sistema.

Postcondiciones: Establece cada cuantos días y a qué hora se actualizará el sistema automáticamente.

Escenario principal:

1. El Administrador solicita modificar los valores de configuración de la actualización.
2. El sistema pide los nuevos datos (Hora, Minutos y Numero de días).
3. El Administrador indica los valores.
4. El sistema comprueba que sean correctos y lo actualiza.

Flujos alternativos:

- *a. En cualquier momento el Administrador cancela la operación.
 1. El sistema cancela la operación.
- 4a. No son valores correctos.
 1. El sistema lo notifica y reinicia la operación.

5.3.26. Especificación del Caso de Uso: Listar Cuentas Nuevas

Caso de Uso: LISTAR CUENTAS NUEVAS

Actores Principales: Adminstrador

Actores Secundarios: —

Precondiciones: El Administrador está logueado.

Postcondiciones: Se muestra una lista con las cuentas registradas que aun no están aceptadas ni rechazadas.

Escenario principal:

1. El Administrador pide al sistema la lista de las Cuentas Nuevas que existen en el sistema.
2. El sistema proporciona la lista.

Flujos alternativos:

- 2a. No existen Cuentas Nuevas en el sistema:
 1. El sistema proporcionará una lista vacía.

5.3.27. Especificación del Caso de Uso: Listar Copias de Seguridad

Caso de Uso: LISTAR COPIAS DE SEGURIDAD

Actores Principales: Administrador

Actores Secundarios: —

Precondiciones: El Administrador está logueado.

Postcondiciones: Se muestra una lista con las cuentas registradas que aun no están aceptadas ni rechazadas.

Escenario principal:

1. El Administrador pide al sistema la lista de las copias de seguridad que existen en el sistema.
2. El sistema proporciona la lista.

Flujos alternativos:

- 2a. No existen copias de seguridad en el sistema:
 1. El sistema proporcionará una lista vacía.

5.3.28. Especificación del Caso de Uso: Crear Copia de Seguridad

Caso de Uso: CREAR COPIA DE SEGURIDAD

Actores Principales: Administrador

Actores Secundarios: —

Precondiciones: El Administrador está logueado en el sistema y hace más de una hora que no se ha hecho otra copia de seguridad.

Postcondiciones: Se almacena en el sistema una copia de seguridad de la base de datos.

Escenario principal:

1. Include: Listar Copias de Seguridad.
2. El Administrador solicita crear una nueva copia de seguridad.
3. El sistema la crea y la almacena.

Flujos alternativos:

- *a. En cualquier momento el Administrador cancela la operación.
 1. El sistema cancela la operación.

5.3.29. Especificación del Caso de Uso: Borrar Copia de Seguridad

Caso de Uso: BORRAR COPIA DE SEGURIDAD

Actores Principales: Administrador

Actores Secundarios: —

Precondiciones: El Administrador está logueado en el sistema.

Postcondiciones: Se borra del sistema una copia de seguridad de la base de datos.

Escenario principal:

1. Include: Listar Copias de Seguridad.
2. El Administrador selecciona una copia de seguridad de la lista y solicita borrarla del sistema.
3. El sistema la elimina.

Flujos alternativos:

- *a. En cualquier momento el Administrador cancela la operación.
 1. El sistema cancela la operación.

5.3.30. Especificación del Caso de Uso: Descargar Copia de Seguridad

Caso de Uso: DESCARGAR COPIA DE SEGURIDAD

Actores Principales: Administrador

Actores Secundarios: —

Precondiciones: El Administrador está logueado en el sistema.

Postcondiciones: Se borra del sistema una copia de seguridad de la base de datos.

Escenario principal:

1. Include: Listar Copias de Seguridad.
2. El Administrador selecciona una copia de seguridad de la lista y solicita descargarla del sistema.
3. El sistema ofrece al Administrador la descarga del archivo.
4. El Administrador acepta la descarga.

Flujos alternativos:

- *a. En cualquier momento el Administrador cancela la operación.
 1. El sistema cancela la operación.

5.3.31. Especificación del Caso de Uso: Login Administrador

Caso de Uso: LOGIN ADMINISTRADOR

Actores Principales: Administrador

Actores Secundarios: —

Precondiciones: El Administrador no está logueado.

Postcondiciones: Se inicia una sesión de Administrador en el sistema.

Escenario principal:

1. El Administrador solicita iniciar sesión.
2. El sistema pide al Administrador que introduzca Login y Pass.
3. El Administrador introduce los datos.
4. EL sistema comprueba que son correctos e inicia la sesión de Alumno.

Flujos alternativo:

- *a. En cualquier momento el Administrador cancela la operación:
 1. El sistema cancela la operación.
- 4a. Los datos no son correctos:
 1. El sistema lo notifica y reinicia la operación.

5.3.32. Especificación del Caso de Uso: Cambiar Contraseña Administrador

Caso de Uso: CAMBIAR CONTRASEÑA ADMINISTRADOR

Actores Principales: Administrador

Actores Secundarios: —

Precondiciones: El Administrador está logueado en el sistema.

Postcondiciones: Asigna una nueva contraseña a la cuenta del Administrador.

Escenario principal:

1. El Administrador solicita cambiar al contraseña
2. El sistema pide la contraseña actual y la nueva.
3. El Administrador indica las contraseñas.
4. El sistema comprueba que las contraseñas sean correctas y la actualiza.

Flujos alternativos:

- *a. En cualquier momento el Administrador cancela la operación.
 1. El sistema cancela la operación.
- 4a. La contraseña actual no es correcta.
 1. El sistema lo notifica y reinicia la operación.
- 4b. La nueva contraseña no es correcta.
 1. El sistema lo notifica y reinicia la operación.

5.3.33. Especificación del Caso de Uso: Cerrar Sesión

Caso de Uso: CERRAR SESIÓN

Actores Principales: Administrador, Profesor o Alumno

Actores Secundarios: —

Precondiciones: El usuario está logueado en el sistema.

Postcondiciones: Se cierra la sesión en curso.

Escenario principal:

1. El usuario cierra la sesión.
2. El sistema cierra la sesión y devuelve al usuario a la página inicial.

5.4. Modelo Conceptual de Datos

A continuación se mostrará el **Modelo Conceptual de Datos**. Este modelo toma como punto de partida considerar la existencia de entidades, que representan objetos, personas, etc. del mundo real distinguibles del resto de objetos por sus características o atributos. Con esto queremos mostrar qué información maneja el sistema y, a grandes rasgos, cómo se relaciona entre sí.

5.4.1. Entidades y Atributos

Antes de mostrar el diagrama conceptual de datos, donde quedarán representadas todas las relaciones y características de las entidades que existen en el sistema especificaremos una lista con todas las entidades existentes y los datos que almacenará el sistema sobre ellas.

- Profesor (Nombre, Apellidos, email, login, Confirmado)
- ProfAlum (LOGprof, DNIALum)
- Alumno (Nombre, Apellidos, DNI, Web, email)
- AlumProy (DNI, Nombre)
- Proyecto (Nombre, Web, Blog, SVN, blogXML, Rev, tipo, Prof)
- ProfConfig (NombreProf, NMes, NSemana, BMes, BSemana, ANMes, ANSemana)
- AlumDesarrollador (DNI, Desarrollador)
- Blog (Nombre, Titulo, Fecha)
- Usuario (Login, pass)
- ProyFlags (NombreProy, Actualizando)
- Stats (Rev, NombreProy, Desarrollador, Mensaje, Fecha, NLineasA, NLineasD, NAdd, NDel, NMod, AplA, AplD, AplM, ImgA, ImgD, WebA, WebD, WebM, DocA, DocD)
- Admin (LOGIN, PASS, hora, min, dias)
- ActFlags (ActualTodo, Nombre, RevF, n, t, b)

5.4.2. Diagrama de Clases Conceptuales

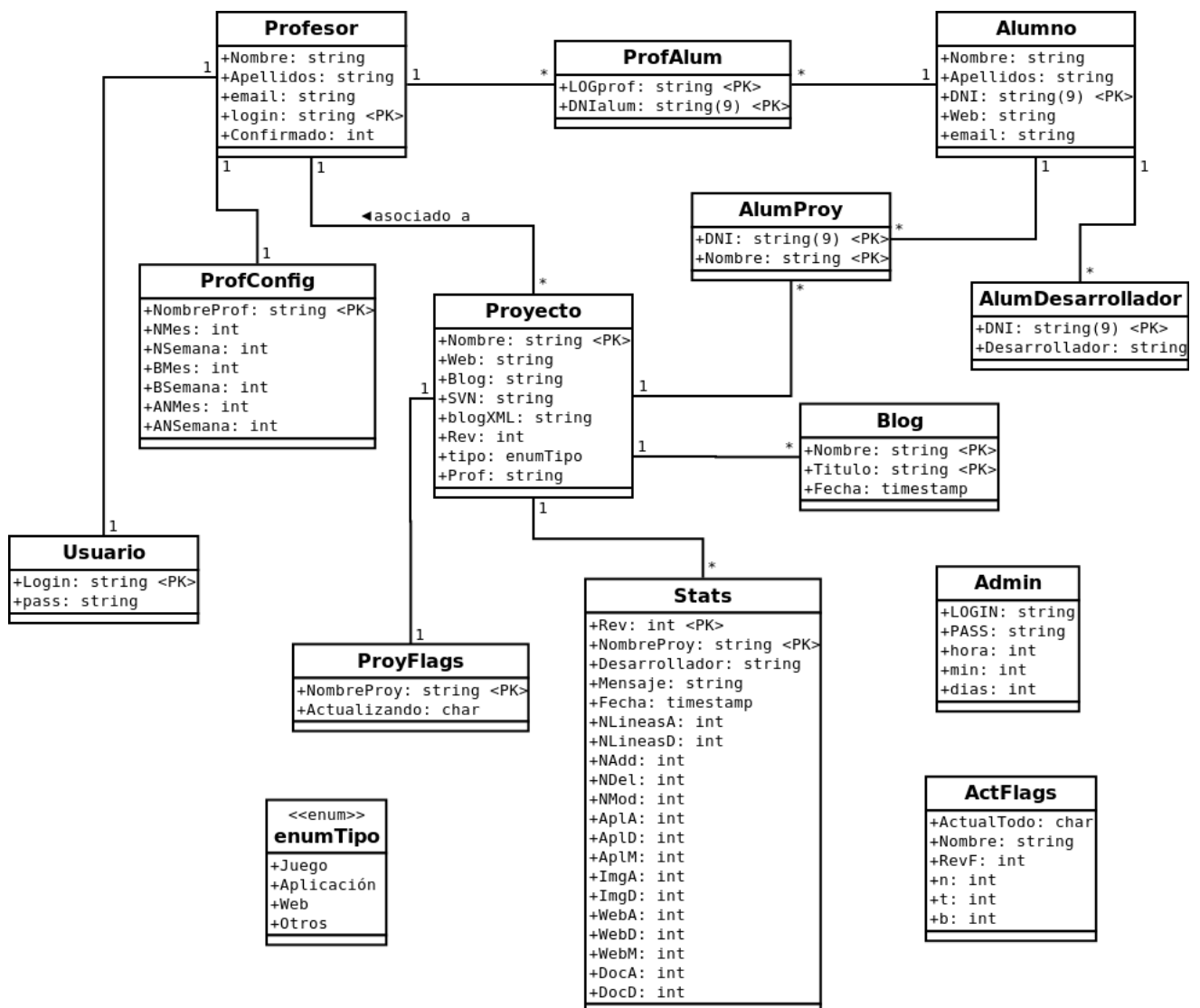


Figura 5.5: Diagrama de Clases Conceptuales.

5.4.3. Restricciones y Reglas

En este apartado, comentamos las características a tener en cuenta en el diagrama conceptual, que no se pueden mostrar a través del gráfico.

- **Restricciones de Clave Externa:** (Profesor, login), (Alumno, DNI), (Proyecto, Nombre).
- Los Alumnos asociados a un Proyecto, han de estar asociados al Profesor del Proyecto.
- Los desarrolladores asociados a los Alumnos (AlumDesarrollador) han de ser desarrolladores de un Proyecto al que están asociados.

5.5. Modelo de Comportamiento del Sistema

En esta sección, explicaremos el comportamiento del sistema en relación a las diferentes funciones e interacciones que se producen. Para ello, haremos uso de los **Diagramas de Secuencia del Sistema** y de los **Contratos de Secuencia del Sistema**.

5.5.1. Diagramas de Secuencia del Sistema

Un diagrama de secuencia es una forma de diagrama de interacción que muestra los objetos como líneas de vida a lo largo de la página y con sus interacciones en el tiempo representadas como mensajes dibujados con flechas desde la línea de vida origen hasta la línea de vida destino. Los diagramas de secuencia son buenos para mostrar qué objetos se comunican con qué otros objetos y qué mensajes disparan esas comunicaciones.

A continuación, en las siguientes apartados, mostramos los diagramas de secuencia del sistema de los distintos casos de uso existentes, agrupando en algunos diagramas varios casos de uso, para no reiterar gráficos que son básicamente iguales.

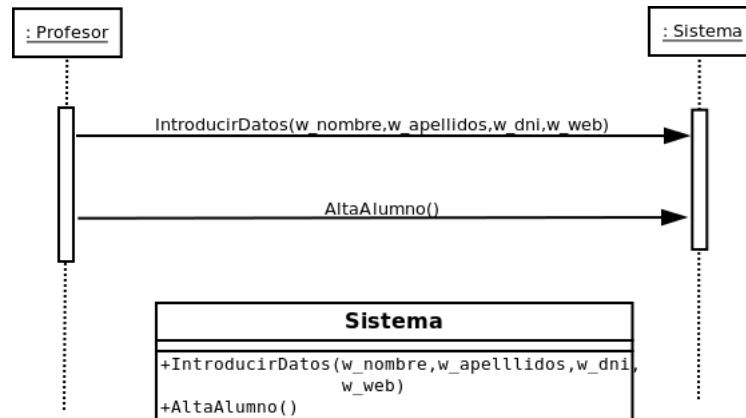
Diagrama de Secuencia del Sistema: Alta Alumno

Figura 5.6: Diagrama de Secuencia del Sistema: Alta Alumno.

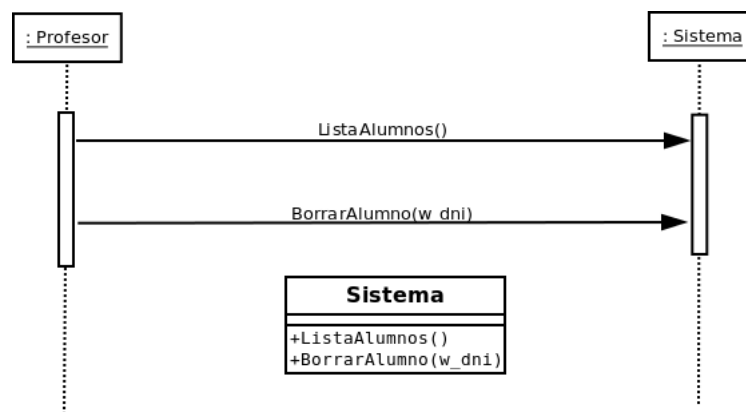
Diagrama de Secuencia del Sistema: Borrar Alumno

Figura 5.7: Diagrama de Secuencia del Sistema: Borrar Alumno.

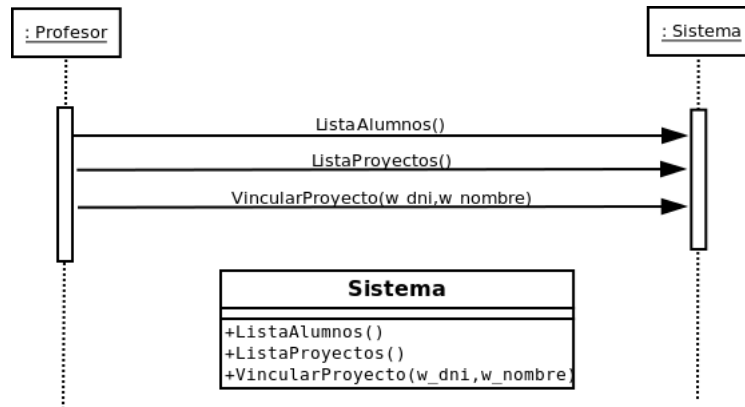
Diagrama de Secuencia del Sistema: Vincular Proyecto

Figura 5.8: Diagrama de Secuencia del Sistema: Vincular Proyecto.

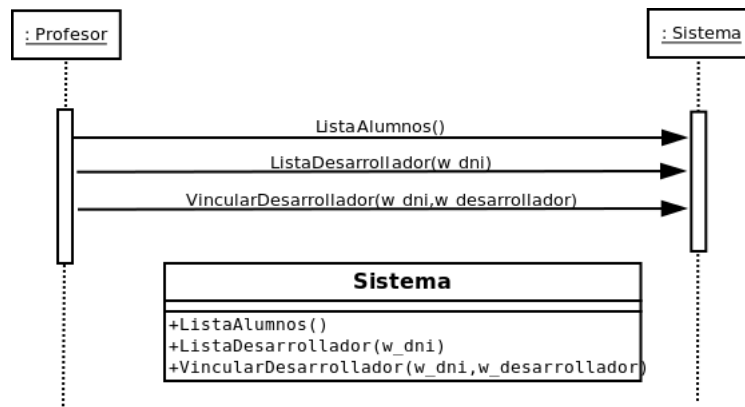
Diagrama de Secuencia del Sistema: Vincular Desarrollador

Figura 5.9: Diagrama de Secuencia del Sistema: Vincular Desarrollador.

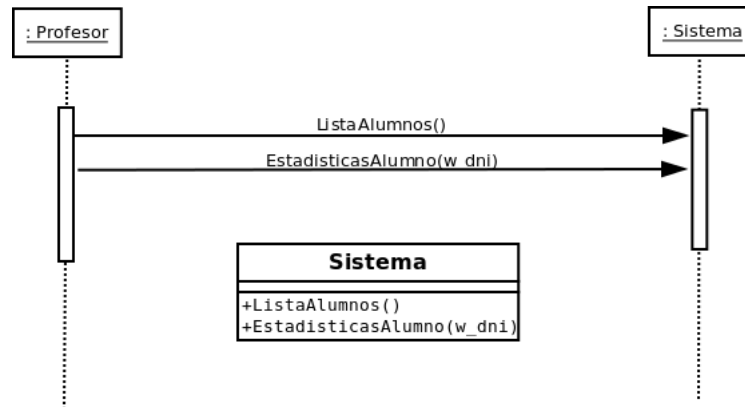
Diagrama de Secuencia del Sistema: Mostrar Estadísticas según Profesor

Figura 5.10: Diagrama de Secuencia del Sistema: Mostrar Estadísticas según Profesor.

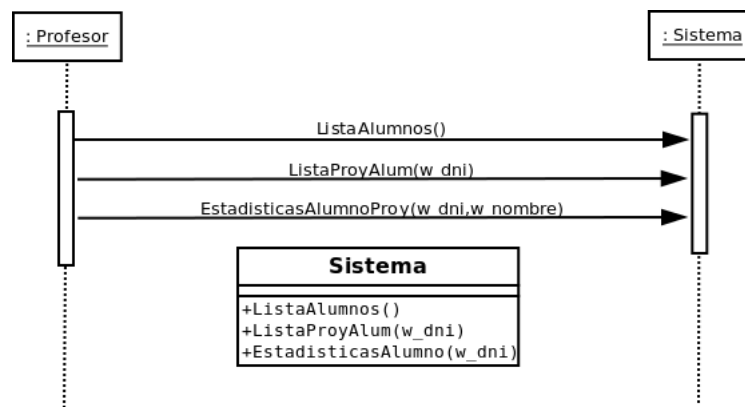
Diagrama de Secuencia del Sistema: Mostrar Estadísticas según Proyecto

Figura 5.11: Diagrama de Secuencia del Sistema: Mostrar Estadísticas según Proyecto.

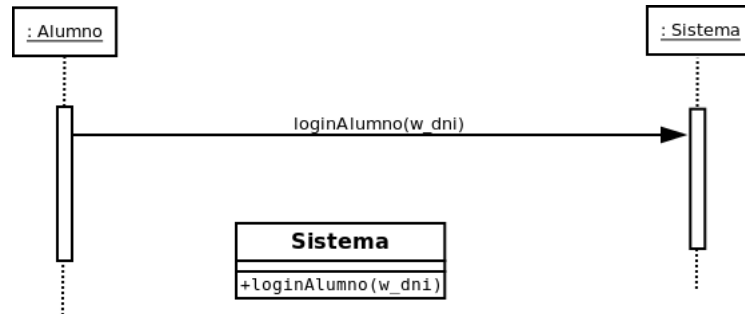
Diagrama de Secuencia del Sistema: Login Alumno

Figura 5.12: Diagrama de Secuencia del Sistema: Login Alumno.

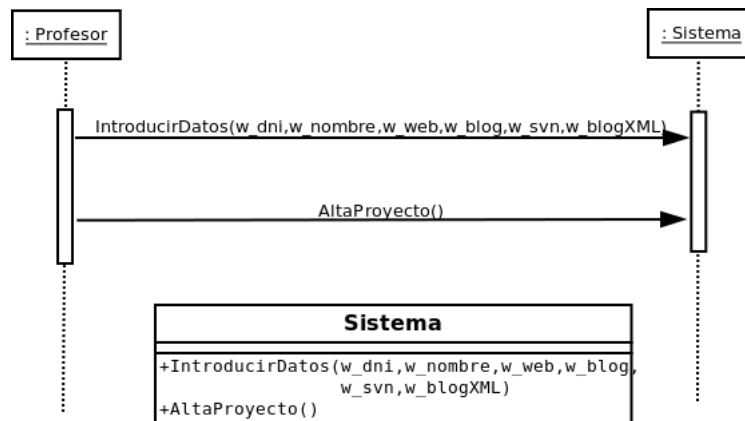
Diagrama de Secuencia del Sistema: Alta Proyecto

Figura 5.13: Diagrama de Secuencia del Sistema: Alta Proyecto.

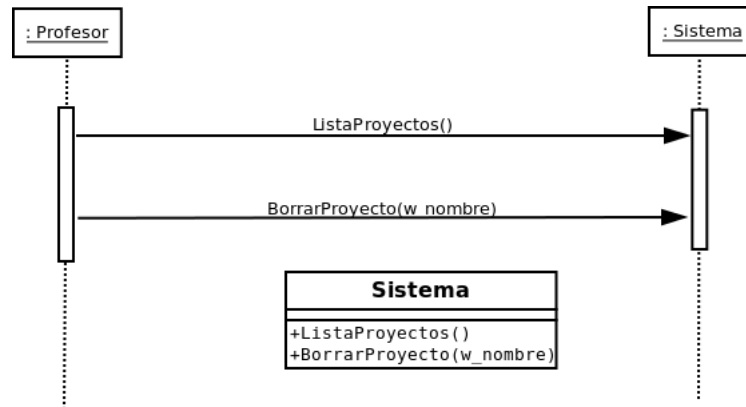
Diagrama de Secuencia del Sistema: Borrar Proyecto

Figura 5.14: Diagrama de Secuencia del Sistema: Borrar Proyecto.

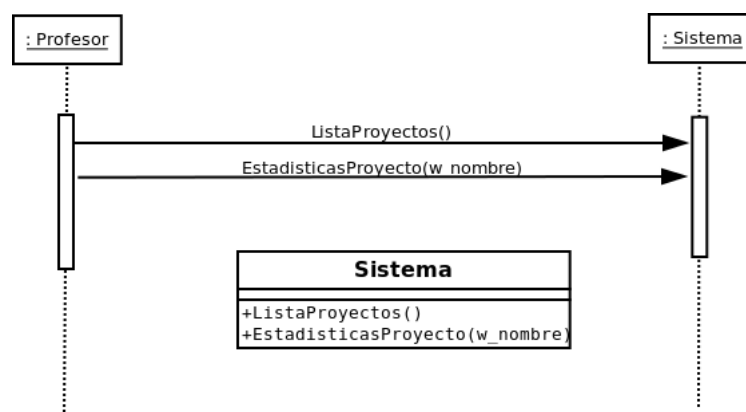
Diagrama de Secuencia del Sistema: Mostrar Estadísticas (Proy)

Figura 5.15: Diagrama de Secuencia del Sistema: Mostrar Estadísticas (Proy).

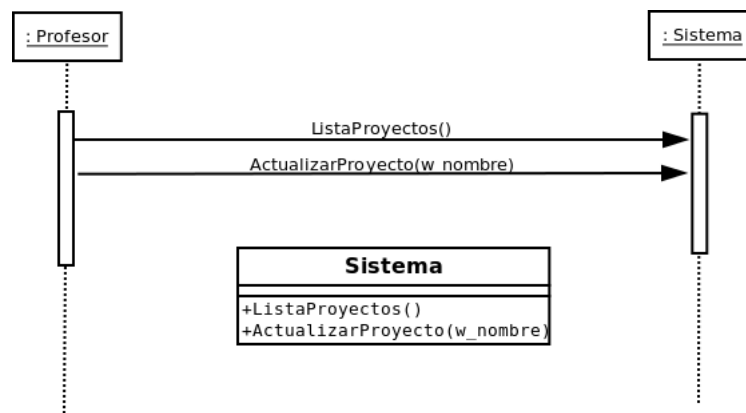
Diagrama de Secuencia del Sistema: Actualizar Proyecto

Figura 5.16: Diagrama de Secuencia del Sistema: Actualizar Proyecto.

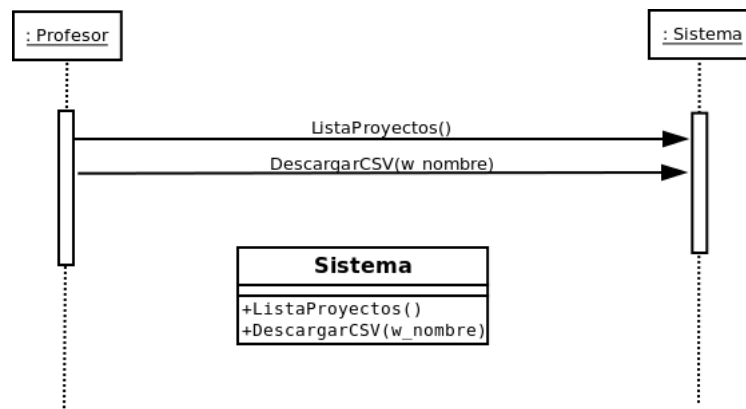
Diagrama de Secuencia del Sistema: Descargar CSV

Figura 5.17: Diagrama de Secuencia del Sistema: Descargar CSV.

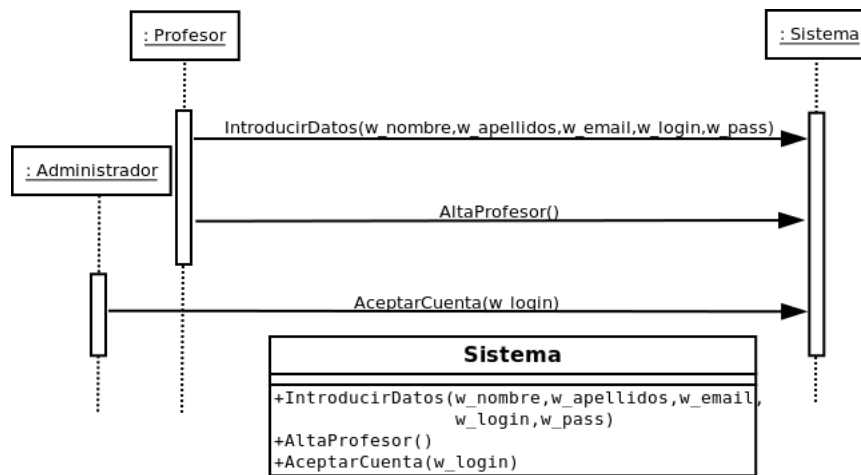
Diagrama de Secuencia del Sistema: Alta Profesor

Figura 5.18: Diagrama de Secuencia del Sistema: Alta Profesor.

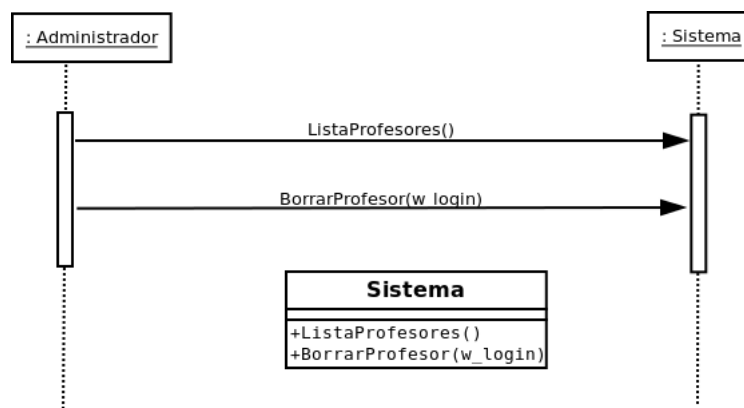
Diagrama de Secuencia del Sistema: Borrar Profesor

Figura 5.19: Diagrama de Secuencia del Sistema: Borrar Profesor.

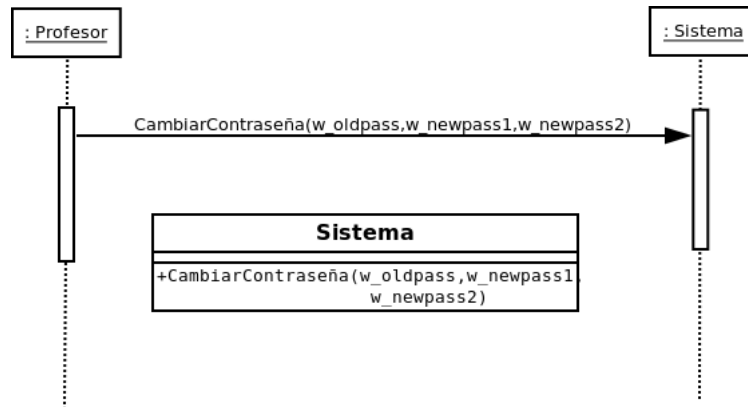
Diagrama de Secuencia del Sistema: Cambiar Contraseña

Figura 5.20: Diagrama de Secuencia del Sistema: Cambiar Contraseña.

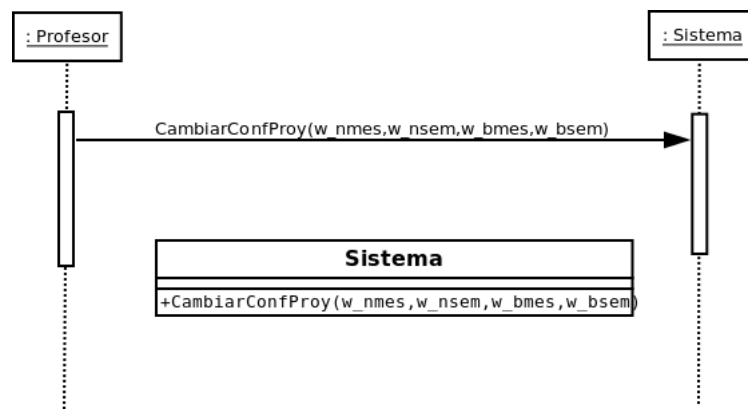
Diagrama de Secuencia del Sistema: Configurar Nivel Proyectos

Figura 5.21: Diagrama de Secuencia del Sistema: Configurar Nivel Proyectos.

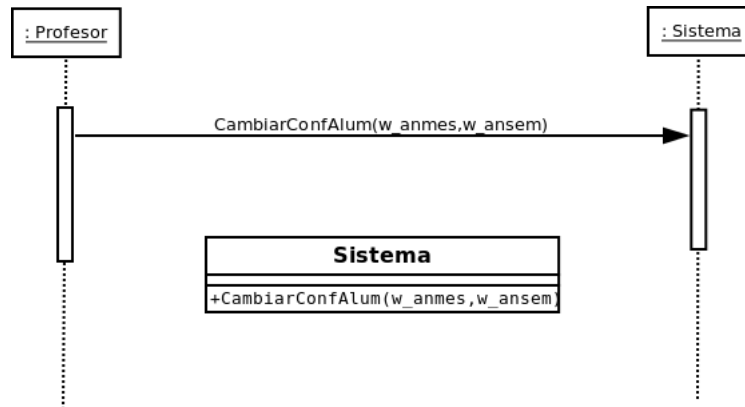
Diagrama de Secuencia del Sistema: Configurar Nivel Alumnos

Figura 5.22: Diagrama de Secuencia del Sistema: Configurar Nivel Alumnos.

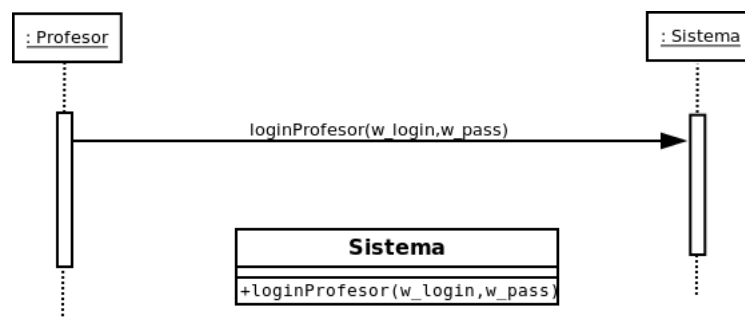
Diagrama de Secuencia del Sistema: Login Profesor

Figura 5.23: Diagrama de Secuencia del Sistema: Login Profesor.

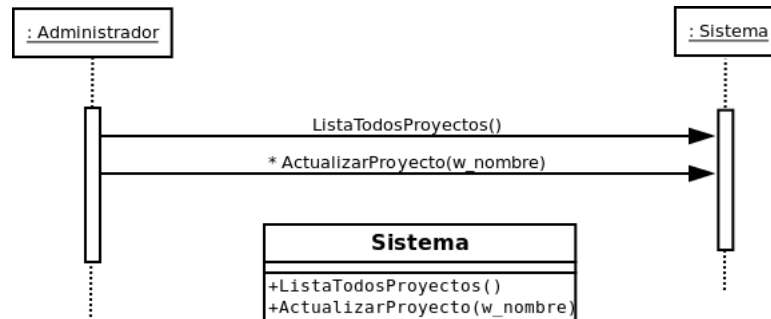
Diagrama de Secuencia del Sistema: Actualizar Todo

Figura 5.24: Diagrama de Secuencia del Sistema: Actualizar Todo.

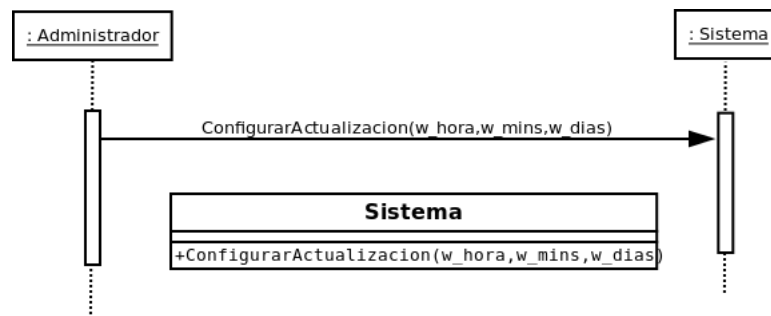
Diagrama de Secuencia del Sistema: Configurar Actualización

Figura 5.25: Diagrama de Secuencia del Sistema: Configurar Actualización.

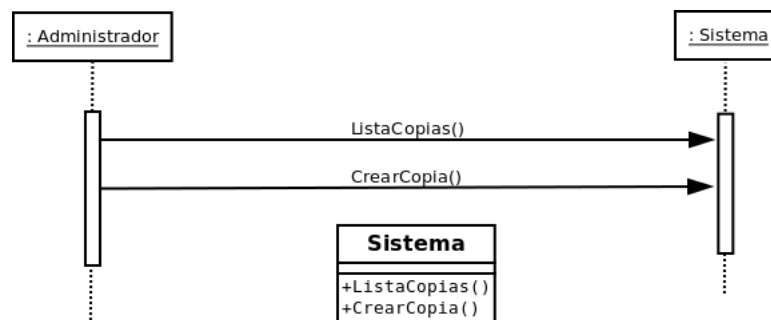
Diagrama de Secuencia del Sistema: Crear Copia de Seguridad

Figura 5.26: Diagrama de Secuencia del Sistema: Crear Copia de Seguridad.

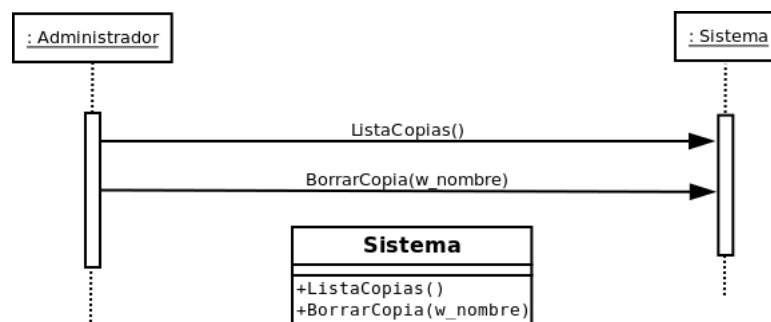
Diagrama de Secuencia del Sistema: Borrar Copia de Seguridad

Figura 5.27: Diagrama de Secuencia del Sistema: Borrar Copia de Seguridad.

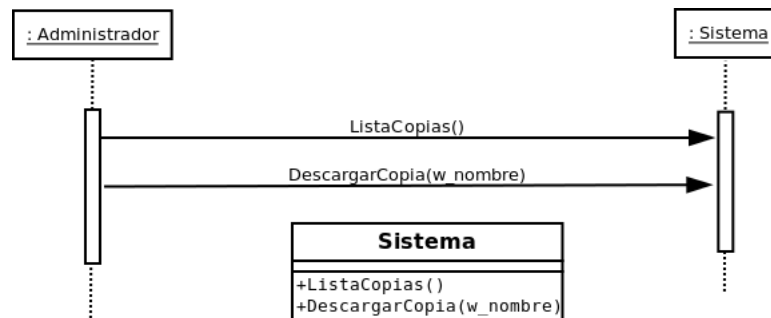
Diagrama de Secuencia del Sistema: Descargar Copia de Seguridad

Figura 5.28: Diagrama de Secuencia del Sistema: Descargar Copia de Seguridad.

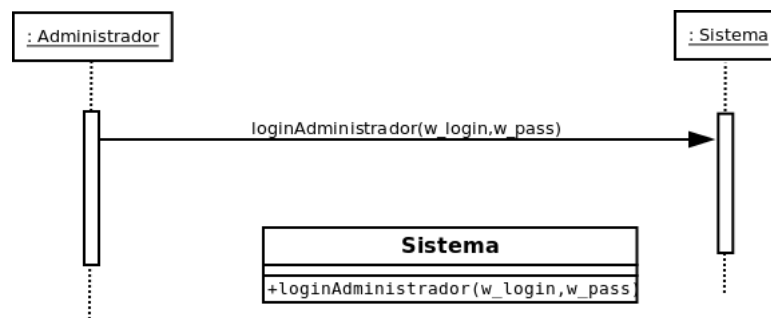
Diagrama de Secuencia del Sistema: Login Administrador

Figura 5.29: Diagrama de Secuencia del Sistema: Login Administrador.

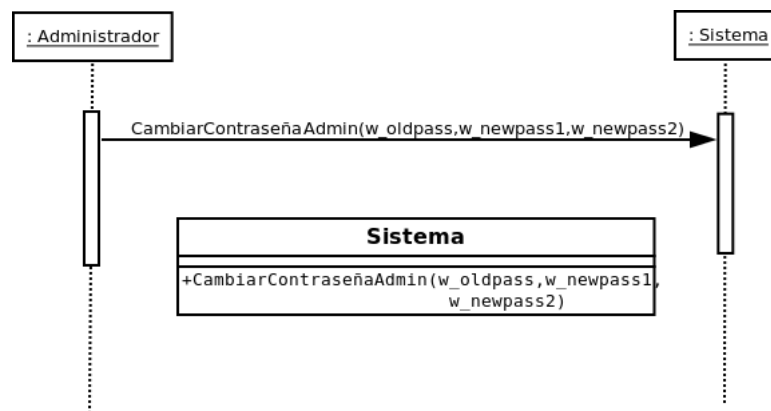
Diagrama de Secuencia del Sistema: Cambiar Contraseña Administrador

Figura 5.30: Diagrama de Secuencia del Sistema: Cambiar Contraseña Administrador.

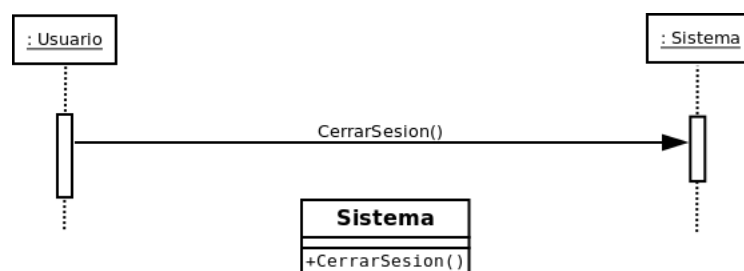
Diagrama de Secuencia del Sistema: Cerrar Sesión

Figura 5.31: Diagrama de Secuencia del Sistema: Cerrar Sesión.

Este diagrama esta representado para cualquier usuario, que como comentamos anteriormente, un usuario se corresponde a **Alumno**, **Profesor** o **Administrador**.

5.5.2. Contratos de las Operaciones del Sistema

A través de los contratos de las operaciones, tendremos constancia de los efectos que producen las distintas operaciones en el sistema.

Contratos de las Operaciones: *IntroducirDatos()*

Operación: *IntroducirDatos(w_nombre, w_apellidos, w_dni, w_web)*

Responsabilidades: Introducir los datos para dar de alta un Alumno.

Referencias Cruzadas: Caso de Uso: Alta Alumno

Precondiciones:

Postcondiciones:

Contratos de las Operaciones: *AltaAlumno()*

Operación: *AltaAlumno()*

Responsabilidades: Registrar un Alumno

Referencias Cruzadas: Caso de Uso: Alta Alumno

Precondiciones:

- No existe un Alumno con $dni == w_dni$ o no está asociado con el Profesor que lo crea.

Postcondiciones:

- Se creó una instancia **C** de Alumno. (*creación de objeto*)
- Se asignó w_dni a **C.dni**, w_nombre a **C.nombre**, $w_apellidos$ a **C.apellidos** y w_web a **C.web**. (*modificación de atributos*).

Contratos de las Operaciones: *ActualizarProyecto()*

Operación: *ActualizarProyecto(w_nombre)*

Responsabilidades: Actualiza los datos del repositorio SVN.

Referencias Cruzadas: Caso de Uso: Actualizar Proyecto

Precondiciones:

Postcondiciones:

- Se creó una instancia **S** de Stats. (*creación de objeto*).
- Se asignó w_nombre a **S.nombreproy**.

Contratos de las Operaciones: *IntroducirDatos()*

Operación: *IntroducirDatos(w_nombre, w_apellidos, w_email, w_login, w_pass)*

Responsabilidades: Registrar un Profesor en el sistema.

Referencias Cruzadas: Caso de Uso: Alta Profesor

Precondiciones:

Postcondiciones:

Contratos de las Operaciones: *AltaAlumno()*

Operación: *AltaAlumno()*

Responsabilidades: Registrar un Profesor en el sistema.

Referencias Cruzadas: Caso de Uso: Alta Profesor

Precondiciones:

- No existe un Profesor con login==w_login.

Postcondiciones:

- Se creó una instancia **P** de Profesor. (*creación de objeto*)
- Se asignó w_login a **P.login**, w_nombre a **P.nombre**, w_apellidos a **P.apellidos**, w_email a **P.email**, w_pass a **P.pass** y 0 a **P.confirmado**. (*modificación de atributos*).

Contratos de las Operaciones: *AceptarCuenta()*

Operación: *AceptarCuenta(w_login)*

Responsabilidades: Acepta la cuenta de un Profesor.

Referencias Cruzadas: Caso de Uso: Alta Profesor.

Precondiciones:

- Existe un Profesor **P** con cuenta sin confirmar.

Postcondiciones:

- Se asignó el valor 1 a **P.confirmado**. (*modificación de atributos*).

Contratos de las Operaciones: *IntroducirDatos()*

Operación: *IntroducirDatos(w_dni, w_nombre, w_web, w_blog, w_svn, w_blogxml)*

Responsabilidades: Registrar un Proyecto en el sistema.

Referencias Cruzadas: Caso de Uso: Alta Proyecto

Precondiciones:

Postcondiciones:

Contratos de las Operaciones: *AltaProyecto()*

Operación: *AltaProyecto()*

Responsabilidades: Registrar un Proyecto en el sistema.

Referencias Cruzadas: Caso de Uso: Alta Proyecto

Precondiciones:

- No existe un Proyecto con nombre==w_nombre.

Postcondiciones:

- Se creó una instancia **P** de Proyecto. (*creación de objeto*)
- Se creó una instancia **A** de AlumProy. (*creación de objeto*)
- Se asignó w_nombre a **P.nombre**, w_web a **P.web**, w_blog a **P.blog**, w_svn a **P.svn**, w_blogxml a **P.blogxml**. (*modificación de atributos*).
- Se asignó w_nombre a **A.nombre**, w_dni a **A.dni**. (*modificación de atributos*).

Contratos de las Operaciones: *BorrarAlumno()*

Operación: *BorrarAlumno(w_dni)*

Responsabilidades: Elimina un Alumno del sistema.

Referencias Cruzadas: Caso de Uso: Borrar Alumno.

Precondiciones:

- Existe un Alumno **A** con dni==w_dni.

Postcondiciones:

- Se eliminó al Alumno **A** del sistema. (*eliminación de objeto*).
- Se eliminó las instancias de AlumProy con dni==w_dni. (*eliminación de objeto*).

Contratos de las Operaciones: *BorrarProfesor()***Operación:** *BorrarProfesor(w_login)***Responsabilidades:** Elimina un Profesor del sistema.**Referencias Cruzadas:** Caso de Uso: Borrar Profesor.**Precondiciones:**

- Existe un Profesor con login==w_login.

Postcondiciones:

- Se eliminó los objetos Proyecto **P** con prof==w_login. (*Eliminación de objeto*).
- Se eliminó los objetos Alumnos **A** con dni==PA.dni, donde **PA** son instancias de ProfAlum con login==w_login. (*Eliminación de asociación*).
- Se eliminó los objetos ProyAlum **PY** con dni==PA.dni, donde **PA** son instancias de ProfAlum con login==w_login. (*Eliminación de asociación*).
- Se eliminó los objetos ProyAlum **PY** con nombre==P.nombre, donde **P** son instancias de Proyecto con prof==w_login. (*Eliminación de asociación*).
- Se eliminó los objetos ProfAlum **PA** con logprof==w_login. (*Eliminación de asociación*).

Contratos de las Operaciones: *BorrarProyecto()***Operación:** *BorrarProyecto(w_nombre)***Responsabilidades:** Elimina un Proyecto del sistema.**Referencias Cruzadas:** Caso de Uso: Borrar Proyecto.**Precondiciones:**

- Existe un Proyecto **P** con nombre==w_nombre.

Postcondiciones:

- Se eliminó al Proyecto **P** del sistema. (*Eliminación de objeto*).
- Se eliminó las instancias de AlumProy con nombre==w_nombre. (*Eliminación de objeto*).

Contratos de las Operaciones: *CambiarContraseña()*

Operación: *CambiarContraseña(w_oldpass, w_newpass1, w_newpass2)*

Responsabilidades: Cambia la contraseña del Profesor.

Referencias Cruzadas: Caso de Uso: Cambiar Contraseña.

Precondiciones:

- Existe un objeto Profesor **P** con $pass=w_oldpass$.
- $w_newpass1==w_newpass2$.

Postcondiciones:

- Se asignó $w_newpass1$ a **P.pass**. (*modificación de atributos*).

Contratos de las Operaciones: *CambiarContraseñaAdmin()*

Operación: *CambiarContraseñaAdmin(w_oldpass, w_newpass1, w_newpass2)*

Responsabilidades: Cambia la contraseña del Administrador.

Referencias Cruzadas: Caso de Uso: Cambiar Contraseña Administrador.

Precondiciones:

- Existe un objeto Admin **A** con $pass=w_oldpass$.
- $w_newpass1==w_newpass2$.

Postcondiciones:

- Se asignó $w_newpass1$ a **A.pass**. (*modificación de atributos*).

Contratos de las Operaciones: *ConfigurarActualizacion()*

Operación: *ConfigurarActualizacion(w_hora, w_mins, w_dias)*

Responsabilidades: Modifica los datos de la actualización automática.

Referencias Cruzadas: Caso de Uso: Configurar Actualización.

Precondiciones:

- Existe un objeto Admin **A** con $pass=w_oldpass$.

Postcondiciones:

- Se asignó w_hora a **A.hora**, w_mins a **A.min** y w_dias a **A.dias**. (*modificación de atributos*).
- Se modifica el archivo crontab con los nuevos valores.

Contratos de las Operaciones: *CambiarConfAlum()***Operación:** *CambiarConfAlum(w_anmes, w_ansem)***Responsabilidades:** Modifica la configuración de nivel de Alumnos.**Referencias Cruzadas:** Caso de Uso: Configurar Nivel de Alumnos**Precondiciones:**

- Existe un Profesor **P**.

Postcondiciones:

- Se modifica el objeto ProfConfig **C** asociado a **P**, C.anmes==w_anmes y C.ansem==w_ansem. (*modificación de atributos*).

Contratos de las Operaciones: *loginAdministrador()***Operación:** *loginAdministrador(w_login, w_pass)***Responsabilidades:** Crea una sesión de Administrador.**Referencias Cruzadas:** Caso de Uso: Login Administrador**Precondiciones:**

- Existe un Administrador **A** con A.login==w_login y A.pass==w_pass.

Postcondiciones:

- Se crea una sesión de Administrador en el sistema. (*modificación de atributos*).

Contratos de las Operaciones: *loginAlumno()***Operación:** *loginAlumno(w_dni)***Responsabilidades:** Crea una sesión de Alumno.**Referencias Cruzadas:** Caso de Uso: Login Alumno**Precondiciones:**

- Existe un Alumno **A** con A.dni==w_dni.

Postcondiciones:

- Se crea una sesión de Alumno en el sistema. (*modificación de atributos*).

Contratos de las Operaciones: *loginProfesor()*

Operación: *loginProfesor(w_login, w_pass)*

Responsabilidades: Crea una sesión de Profesor.

Referencias Cruzadas: Caso de Uso: Login Administrador

Precondiciones:

- Existe un Usuario **U** con $U.login == w_login$ y $U.pass == w_pass$.

Postcondiciones:

- Se crea una sesión de Profesor en el sistema. (*modificación de atributos*).

Contratos de las Operaciones: *NuevaContraseña()*

Operación: *NuevaContraseña(w_oldpass, w_newpass1, w_newpass2)*

Responsabilidades: Cambia la contraseña del Profesor.

Referencias Cruzadas: Caso de Uso: Cambiar Contraseña Administrador.

Precondiciones:

- Existe un objeto Usuario **U** con $U.pass = w_oldpass$ y $U.login == w_login$.
- $w_newpass1 == w_newpass2$.

Postcondiciones:

- Se asignó $w_newpass1$ a **U.pass**. (*modificación de atributos*).

Contratos de las Operaciones: *VincularDesarrollador()*

Operación: *VincularDesarrollador(w_dni, w_desarrollador)*

Responsabilidades: Crea una relación entre un Alumno y un Desarrollador de un Proyecto.

Referencias Cruzadas: Caso de Uso: Vincular Desarrollador.

Precondiciones:

- Existe un objeto Proyecto **P**.
- Existe un objeto ProyAlum **PY** con $PY.dni == w_dni$ y $PY.nombre == P.nombre$.

Postcondiciones:

- Se creó una instancia de AlumDesarrollador **AD** con $AD.dni == w_dni$ y $AD.desarrollador == w_desarrollador$. (*creación de enlace*).

Contratos de las Operaciones: VincularProyecto()

Operación: *VincularProyecto(w_dni,w_nombre)*

Responsabilidades: Crea una relación entre un Alumno y un Proyecto.

Referencias Cruzadas: Caso de Uso: Vincular Proyecto.

Precondiciones:

- Existe un objeto Proyecto **P** con P.nombre==w_nombre.
- Existe un objeto Alumno **A** con A.dni==w_dni.

Postcondiciones:

- Se creó una instancia de AlumProy **PY** con PY.dni==w_dni y PY.nombre==w_nombre.
(*creación de enlace*).

Operaciones Irrelevantes

Las operaciones de los casos de uso, listados a continuación, son irrelevantes porque no producen cambios en el modelo. Por tanto, no realizaremos sus contratos:

- Lista Proyectos
- Lista Todos los Proyectos
- Lista Alumnos
- Lista Copias
- Borrar Copia
- Lista Profesores
- Crear Copia
- Descargar Copia
- Descargar CSV
- Estadísticas Alumno
- Estadísticas Proyecto
- Lista Desarrollador

Capítulo 6

Diseño del Sistema

6.1. Introducción

El diseño de aplicaciones software está en constante evolución desde sus inicios. En los primeros diseños de aplicaciones monolíticas se mezclaban la lógica de procesamiento, la interfaz de usuario y la información almacenada. Más tarde, el diseño de aplicaciones evolucionó hacia un nuevo modelo, las aplicaciones con arquitectura cliente/servidor, donde se repartía la carga computacional en dos partes independientes: una parte denominada cliente y la otra servidor, pero sin establecer una delimitación clara de funciones.

Actualmente, el modelo cliente-servidor ha seguido evolucionando hacia una arquitectura donde la carga se divide en tres partes con un reparto claro de funciones: una **capa de presentación**, una **capa de cálculo o procesamiento** y una **capa para el almacenamiento**.

Esta evolución de la arquitectura cliente-servidor a la arquitectura en 3 capas no se ha detenido, sino que ha seguido especializando el reparto de funciones, dando lugar a lo que se conoce como *arquitectura en n-capas*.

6.2. Arquitectura Modelo-Vista-Controlador

La aplicación estDV está desarrollada usando CodeIgniter. CodeIgniter es un conjunto de herramientas para personas que construyen su aplicación web usando PHP. Su objetivo es permitirle desarrollar proyectos mucho más rápido de lo que podría si lo escribiese desde cero, proveyendole un rico juego de librerías para tareas comunmente necesarias, así como una interface simple y estructura lógica para acceder a esas librerías. Este framework de php está basado en el patrón de desarrollo Modelo-Vista-Controlador. El patrón Modelo-Vista-Controlador es una aproximación al software que separa la lógica de la aplicación de la presentación.

La arquitectura en 3 capas, adoptada para el diseño de la aplicación que ocupa este proyecto, es la mas básica de las arquitecturas de n-capas. En ella se pueden distinguir los niveles de:

- **Modelo:** El Modelo representa la estructura de datos. Típicamente sus clases de modelo contendrán funciones que lo ayudarán a recuperar, insertar y actualizar información en su base de datos.
- **Vista:** La Vista es la información que es presentada al usuario. La Vista normalmente será una página web, pero en CodeIgniter, una vista también puede ser un fragmento de una página como un encabezado o un pie de página. También puede ser una página RSS, o cualquier otro tipo de "página".
- **Controlador:** El Controlador sirve como un intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para procesar la petición HTTP y generar una página web.

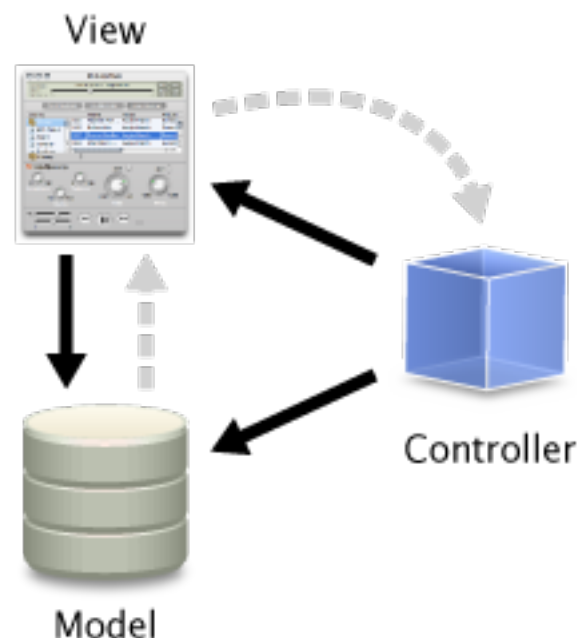


Figura 6.1: Arquitectura Modelo-Vista-Controlador.

6.3. Capa de Vistas

La aplicación estDV está pensada para la comodidad. El gran objetivo, como hemos indicado ya en varias ocasiones, es la facilidad de acceso a los datos (teniendo en cuenta siempre la seguridad), la comodidad de acceso y el que sea agradable a la vista, sin colores ni formas agresivas.

El navegador en el que se ejecute la aplicación hará que la vista cambie levemente, debido a las peculiaridades de cada uno de ellos. Aun así la visión general y la estructura del diseño visual no variará. Esto lo tenemos asegurado gracias a la validación proporcionada por W3C, que indica que la aplicación cumple los estándares.

6.4. Capa de Controladores (*Diagramas de Interacción*)

Los controladores están separados según que rol desempeña la acción. Es decir, un alumno accederá al controlador de alumnos y por consiguiente a una zona de la aplicación y con unas opciones distintas a un Profesor o un Administrador.

6.4.1. Diagramas de Interacción: Sección de Alumnos

Un usuario, tras loguearse exitosamente como Alumno, accederá a las siguientes opciones:

Proyectos
Seguimiento

Tabla 6.1: Opciones del Menú de Alumnos

Al seleccionar la opción **Proyectos**, se listará los proyectos a los que está asociado. Las opciones de este apartado son:

Gestionar
Seguimiento

Tabla 6.2: Opciones del Menú Alumnos-Proyectos

Dentro de **Gestionar**, se mostrará, además de información sobre el proyecto, algunas opciones:

Actualizar
Seguimiento
Descargar CSV

Tabla 6.3: Opciones del Menú Alumno-Proyectos-Gestionar

Alta Alumno
Alta Proyecto
Mostrar Alumnos
Mostrar Proyectos
Configuración

Tabla 6.4: Opciones del Menú de Profesor

Por otro lado si el usuario fuese un Profesor, tras haberse logueado, accederá a las siguientes opciones:

Cuando se accede a **Mostrar Alumnos**, se mostrará, además de la lista de Alumnos asociados al Profesor, algunas opciones:

Seguimiento
Seguimiento por Proyecto
Mostrar Proyectos
Vincular Proyecto
Vincular Desarrollador
Borrar

Tabla 6.5: Opciones del Menú Profesor-Mostrar Alumnos

Las opciones **Vincular Proyecto** y **Vincular Desarrollador** mostrarán una lista de Proyectos o Desarrolladores respectivamente con la opción de vincular.

Tras acceder a **Mostrar Proyectos** tenemos la opción de gestionar y dentro encontramos las opciones:

En la parte de configuración encontramos tres opciones:

Por último si el usuario logueado es un Administrador, su panel de opciones contendrá:

Modificar
Borrar
Estadísticas
CSV

Tabla 6.6: Opciones del Menú Profesor-Mostrar Proyectos-Gestionar

Proyectos
Alumnos
Datos

Tabla 6.7: Opciones del Menú Profesor-Configuración

Profesores
Alumnos
Proyectos
Nuevas Cuentas
Actualizar
Copia de Seguridad
Configuración

Tabla 6.8: Opciones del Menú de Profesor

Crear Copia
Borrar
Descargar

Tabla 6.9: Opciones del Menú de Profesor

Las opciones **Profesores**, **Alumnos**, **Proyectos** contienen una lista de Profesores, Alumnos o Proyectos respectivamente y la opción de borrar el seleccionado. De igual forma **Nuevas Cuentas** da la opción de aceptar o rechazar cuentas creadas.

La opción **Copia de Seguridad** nos lleva a estas opciones:

La opción **Configuración** nos lleva a estas opciones:

Actualizar
Datos

Tabla 6.10: Opciones del Menú de Profesor

Importante: Para simplificar, la parte gráfica de la interfaz del usuario no la vamos a detallar. Por tanto, vamos a utilizar una única clase representativa de toda la interfaz de usuario implicada en la interacción (Pantalla Menú). Además, vamos a emplear una clase de control para controlar los eventos asociados a la gestión del menú principal según su rol. A esta clase la vamos a llamar, por ejemplo, *Administrador Control*.

Por otro lado, mencionar que, como el diagrama de todos los menús son prácticamente iguales, sólo mostraremos uno de ejemplo.

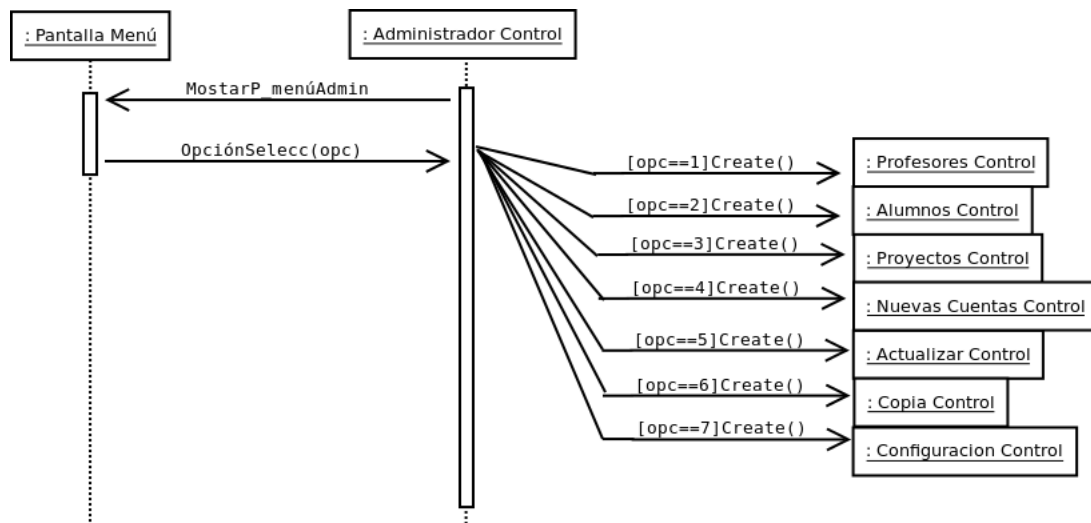


Figura 6.2: Diagrama de Interacción del Menú Principal de Administrador.

Para ejecutar la aplicación, primero se crea un objeto de la clase Sistema Control (*new.SistemaControl*) y luego se ejecutaría una operación de la clase Sistema Control que invorúa las operaciones necesarias para el login del usuario en el sistema. Suponiendo que el login haya sido de Administrador se crearía la clase *Administrador Control* que llamaría a la operación *MostrarP_menúAdmin* que se encarga de mostrar la pantalla del menú de la aplicación (en realidad esta operación invocará a otras operaciones de las clases de la parte de interfaz de usuario para mostrar la pantalla del menú de la aplicación).

A continuación, una vez mostrada la pantalla del menú de opciones, se está a la espera de que se produzca algún evento. En este caso el evento que se puede producir es que el usuario seleccione una determinada opción del menú (*Opción seleccionada(opc)*).

En función de la opción del submenu que el usuario seleccione, se crea un objeto de la clase de control que controla la acción asociada a la opción (caso de uso) seleccionada. **Ejemplo:** si se selecciona la opción Alumnos, se crea un objeto de la clase control Alumnos (*new.Alumnos Control*) y se ejecuta una operación definida en esta clase que le manda un mensaje a la Interfaz de Usuario para que muestre la pantalla de listado de Alumnos.

6.4.2. Diagrama de Interacción: Actualizar Proyecto

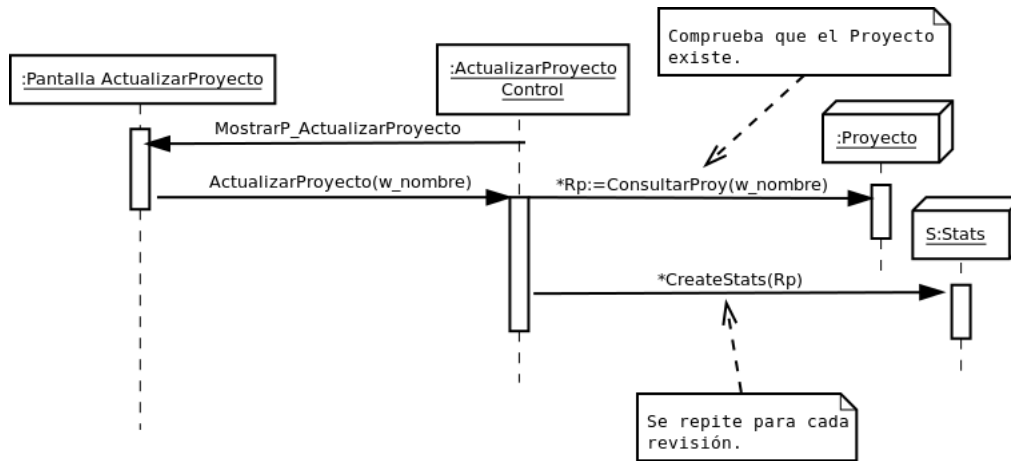


Figura 6.3: Diagrama de Interacción del Caso de Uso: Actualizar Proyecto.

6.4.3. Diagrama de Interacción: Alta Alumno

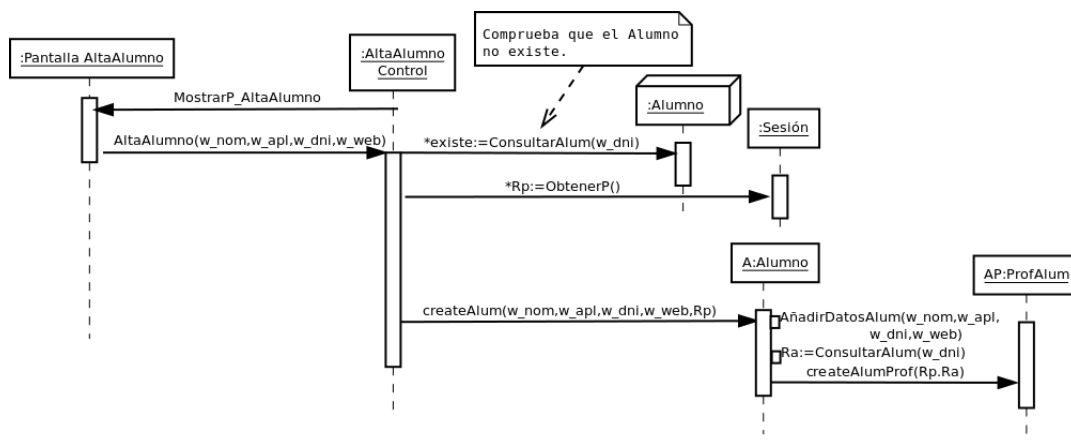


Figura 6.4: Diagrama de Interacción del Caso de Uso: Alta Alumno.

6.4.4. Diagrama de Interacción: Mostrar Estadísticas de Proyecto

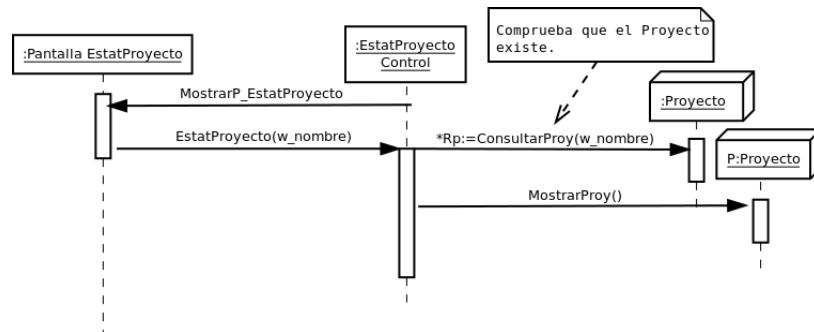


Figura 6.5: Diagrama de Interacción del Caso de Uso: Mostrar Estadísticas de Proyecto.

6.4.5. Diagrama de Interacción: Borrar Proyecto

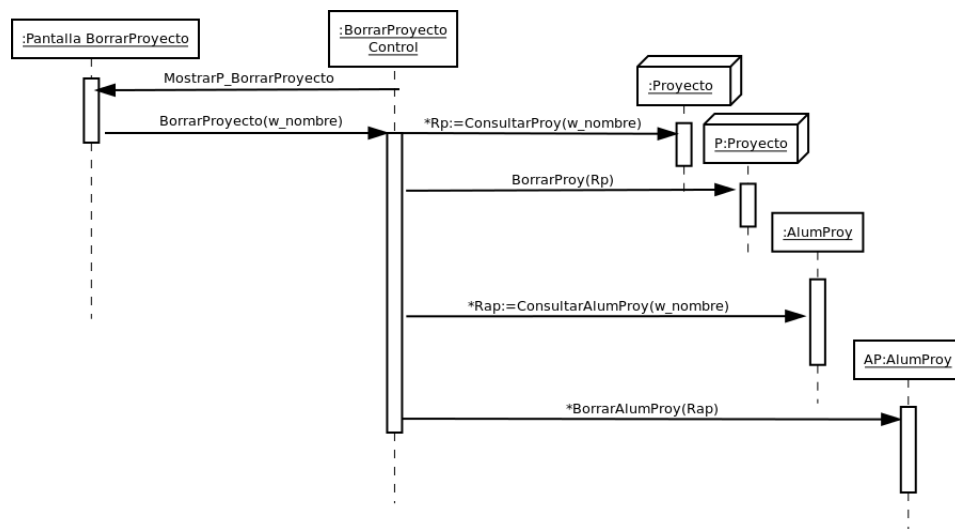


Figura 6.6: Diagrama de Interacción del Caso de Uso: Borrar Proyecto.

6.4.6. Diagrama de Interacción: Configurar Nivel Proyectos

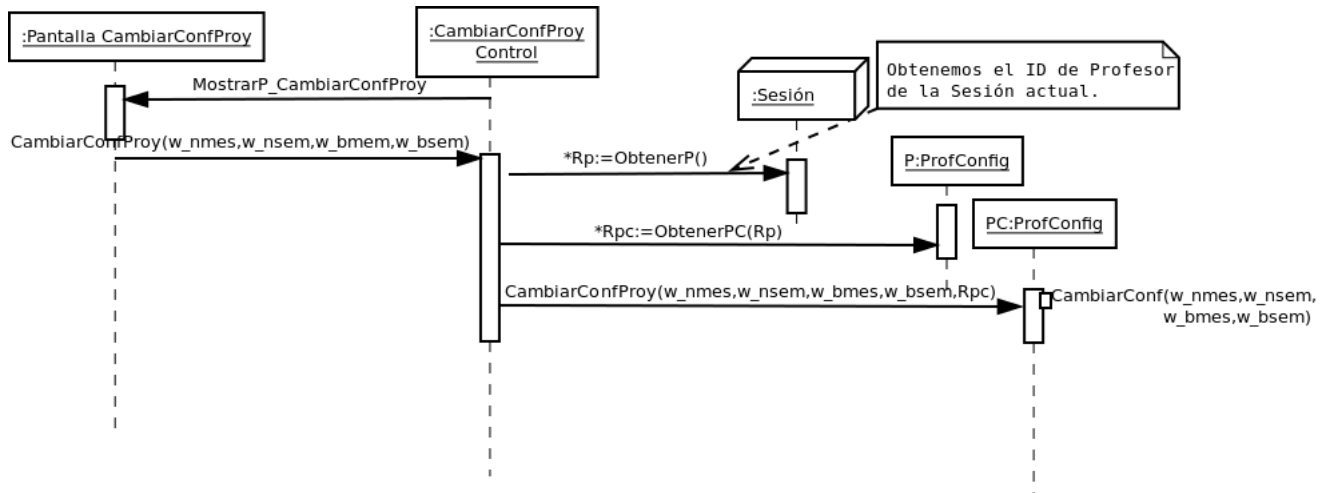


Figura 6.7: Diagrama de Interacción del Caso de Uso: Configurar Nivel Proyectos.

6.4.7. Diagrama de Interacción: Cerrar Sesión

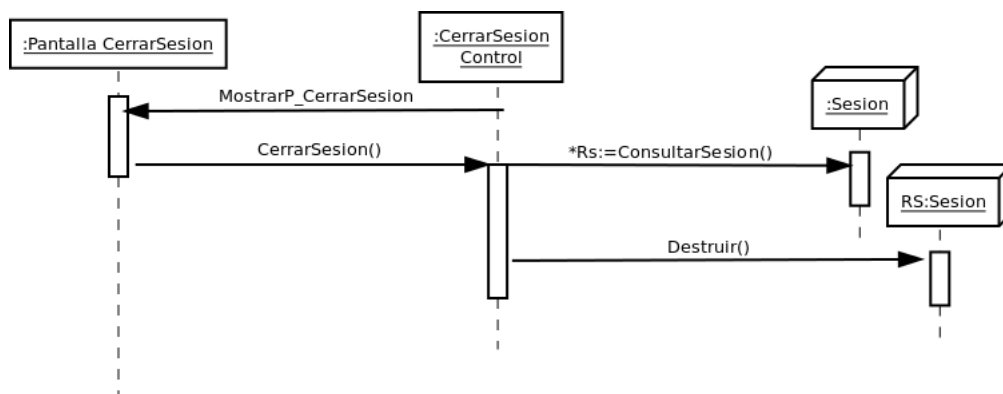


Figura 6.8: Diagrama de Interacción del Caso de Uso: Cerrar Sesión.

6.4.8. Diagrama de Interacción: Crear Copia de Seguridad

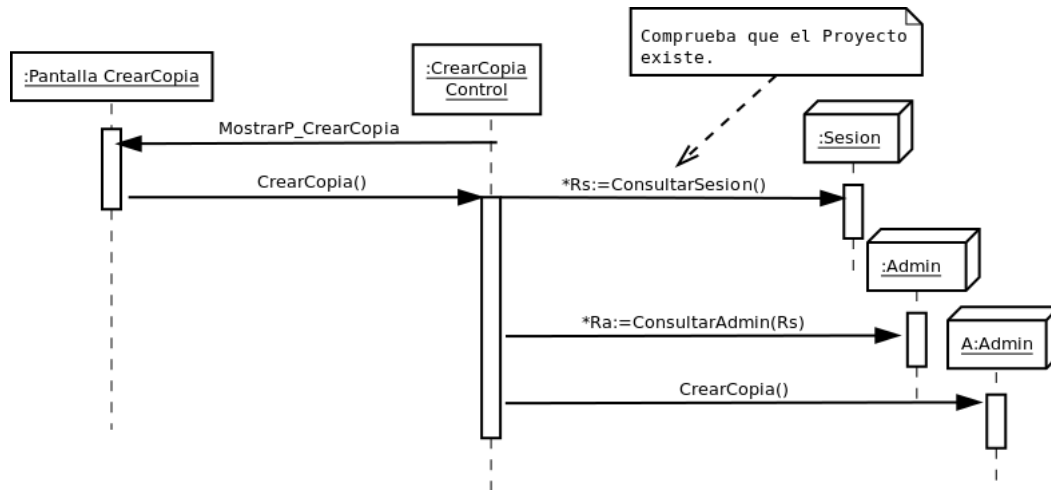


Figura 6.9: Diagrama de Interacción del Caso de Uso: Crear Copia de Seguridad.

6.4.9. Diagrama de Interacción: Descargar CSV

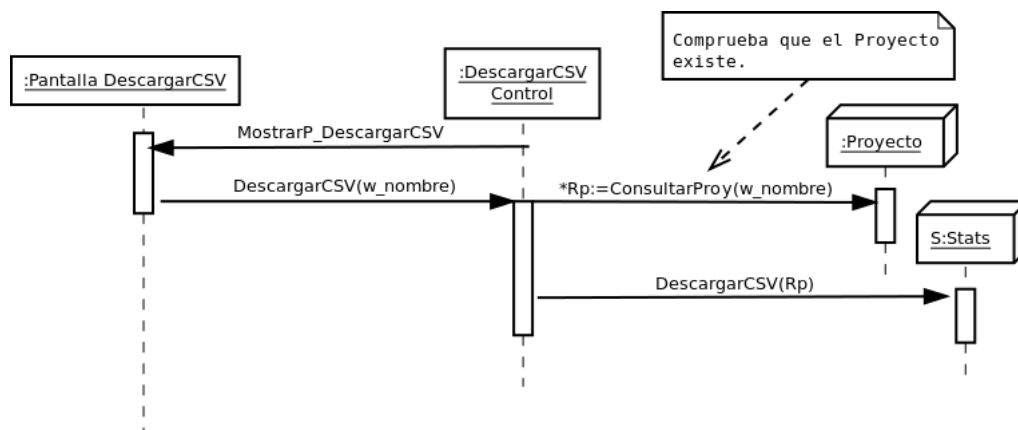


Figura 6.10: Diagrama de Interacción del Caso de Uso: Descargar CSV.

6.4.10. Diagrama de Interacción: Mostrar Estadísticas de Proyecto

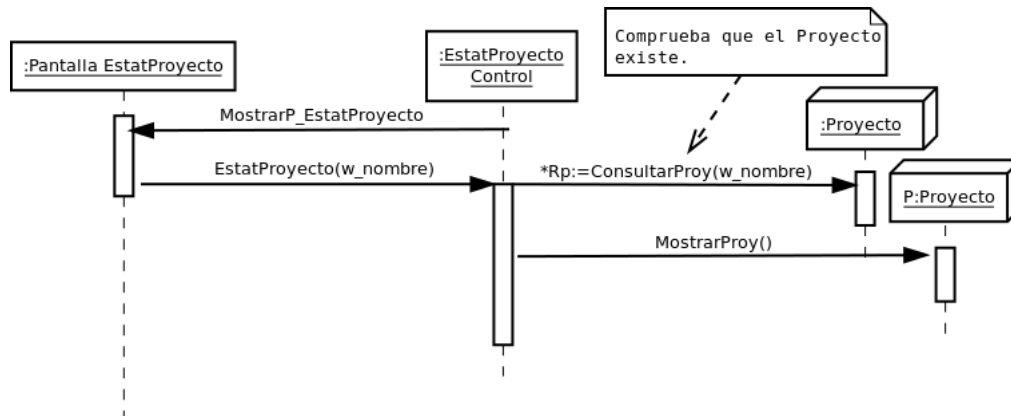


Figura 6.11: Diagrama de Interacción del Caso de Uso: Mostrar Estadísticas de Proyecto.

6.4.11. Diagrama de Interacción: Login Profesor

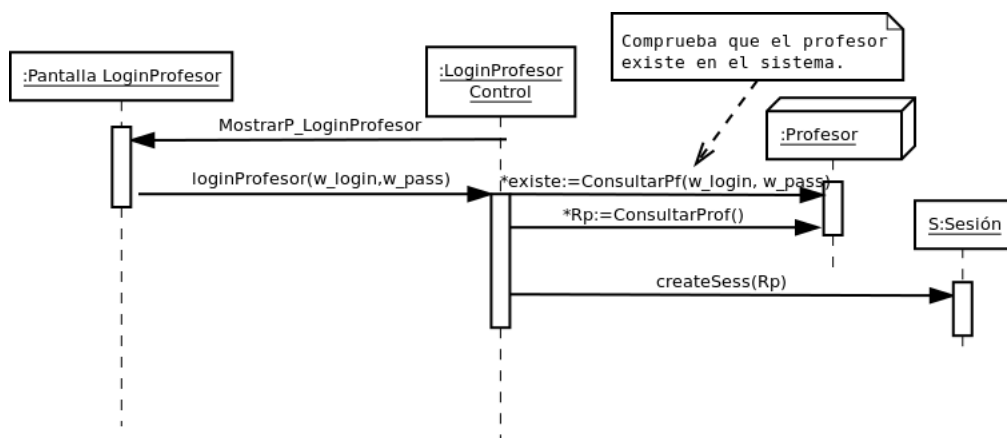


Figura 6.12: Diagrama de Interacción del Caso de Uso: Login Profesor.

6.4.12. Diagrama de Interacción: Vincular Proyecto

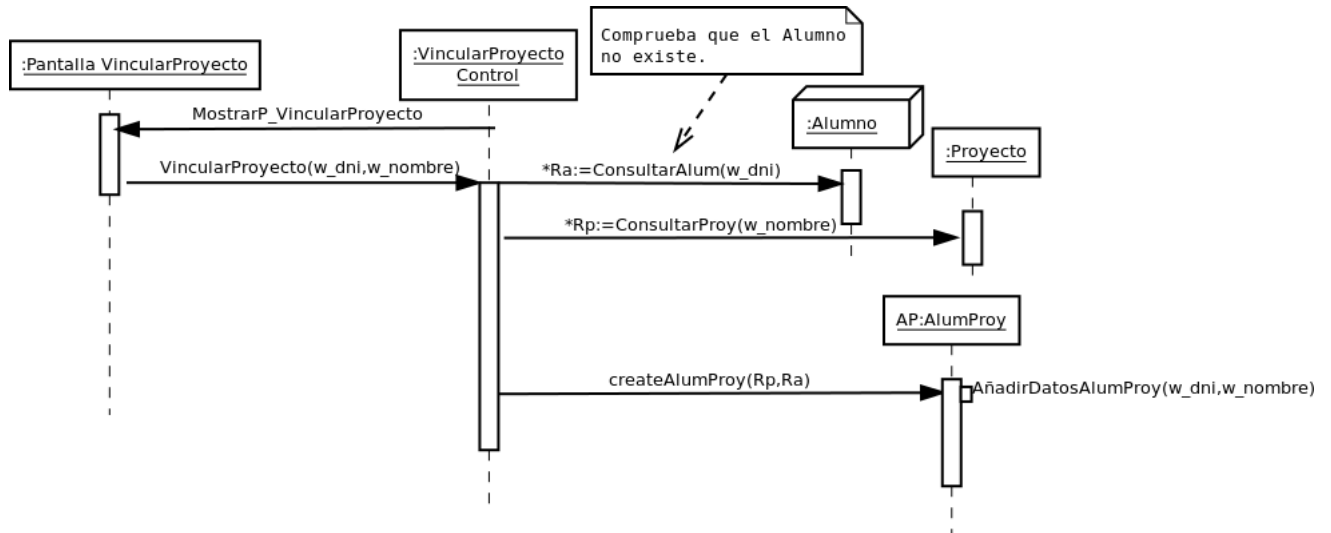


Figura 6.13: Diagrama de Interacción del Caso de Uso: Vincular Proyecto.

6.4.13. Contrato de las Operaciones

Con los diagramas de interacción se obtienen nuevas clases, asociaciones y operaciones. A continuación mostramos algunas de esas clases, asociaciones, y operaciones que surgen, en concreto las relacionadas con los diagramas de interacción descritos arriba.

Clases Nuevas

- Pantalla Menú
- Administrador Control
- Profesores Control
- Alumnos Control
- Proyectos Control
- Nuevas Cuentas Control
- Actualizar Control
- Copia Control
- Configuración Control

- Pantalla ActualizarProyecto
- ActualizarProyecto Control
- Proyecto
- Stats
- Pantalla AltaAlumno
- AltaAlumno Control
- Alumno
- Sesion
- ProfAlum
- Pantalla BorrarProyecto
- BorrarProyecto Control
- AlumProy
- Pantalla CambiarConfProy
- CambiarConfProy Control
- ProfConfig
- Pantalla CerrarSesion
- CerrarSesion Control
- Pantalla CrearCopia
- CrearCopia Control
- Admin
- Pantalla DescargarCSV
- DescargarCSV Control
- Pantalla EstatProyecto
- EstatProyecto Control
- Pantalla LoginProfesor
- LoginProfesor Control
- Pantalla VincularProyecto
- VincularProyecto Control
- (...)

Colaboraciones entre Objetos

- Pantalla Menú colabora con Administrador Control.
- Sistema Control colabora con Profesores Control, Alumnos Control, Proyectos Control, Nuevas Cuentas Control, Actualizar Control, Copia Control y Configuración Control.
- Pantalla ActualizarProyecto colabora con ActualizarProyecto Control.
- ActualizarProyecto Control colabora con Proyecto y Stats.
- Pantalla AltaAlumno colabora con AltaAlumno Control.
- AltaAlumno Control colabora con Alumno, Sesión.
- Alumno colabora con ProfAlum.
- Pantalla BorrarProyecto colabora BorrarProyecto Control.
- BorrarProyecto Control colabora con Proyecto y AlumProy.
- Pantalla CambiarConfProy colabora con CambiarConfProy Control.
- CambiarConfProy Control colabora con Sesión y ProfConfig.
- Pantalla CerrarSesion colabora con CerrarSesion Control.
- CerrarSesion Control colabora con Sesión.
- Pantalla CrearCopia colabora con CrearCopia Control.
- CrearCopia Control colabora con Sesión y Admin.
- Pantalla DescargarCSV colabora con DescargarCSV Control.
- DescargarCSV Control colabora con Proyecto y Stats.
- Pantalla EstatProyecto colabora con EstatProyecto Control.
- EstatProyecto Control colabora con Proyecto.
- Pantalla LoginProfesor colabora con LoginProfesor Control.
- LoginProfesor Control colabora con Profesor y Sesión.
- Pantalla VincularProyecto colabora con VincularProyecto Control.
- VincularProyecto Control colabora con Alumno, Proyecto y AlumProy.
- (...)

Operaciones

■ Clase Pantalla Menú

- Operación *MostrarP_menúAdmin*. Esta operación muestra la pantalla con el menu de Administrador.

■ Clase Administrador Control

- Operación que se ejecuta al hacer *new.AdministradorControl*. Esta operación invoca a la operación *MostrarP_menúAdmin* de la clase Pantalla Menú.
- *OpcionSeleccionada (opc)*. Esta operación crea un objeto de la clase de control correspondiente la opción seleccionada (*new.ClaseControl*).

■ Clase Pantalla ActualizarProyecto

- Operación *MostrarP_ActualizarProyecto*. Esta operación muestra la pantalla de la función Actualizar Proyecto.

■ Clase ActualizarProyecto Control

- Operación que se ejecuta al hacer *new.ActualizarProyectoControl*. Esta operación invoca a la operación *MostrarP_ActualizarProyecto* de la clase Pantalla ActualizarProyecto.
- *ActualizarProyecto(w_nombre)*. Esta operación invoca a la operación **Rp:=ConsultaProy(w_nombre)* de la clase Proyecto y a la operación **CreateStats(Rp)* de la clase Stats.

■ Clase Pantalla AltaAlumno

- Operación *MostrarP_AltaAlumno*. Esta operación muestra la pantalla de la función Alta Alumno.

■ Clase AltaAlumno Control

- Operación que se ejecuta al hacer *new.AltAlumnoControl*. Esta operación invoca a la operación *MostrarP_AltaAlumno* de la clase Pantalla AltaAlumno.
- *AltaAlumno(w_nom,w_apl,w_dni,w_web)*. Esta operación invoca a la operación **existe:=ConsultarAlum(w_dni)* de la clase Alumno, a la operación **Rp:=ObtenerP()* de la clase Sesión y a la operación *createAlum(w_nom,w_apl,w_dni,w_web)* de la clase Alumno.

■ Clase Alumno

- *createAlum(w_nom,w_apl,w_dni,w_web)*. Esta operación invoca a *AñadirDatos-Alum(w_nom,w_apl,w_dni,w_web)* de la clase Alumno, *Ra:=ConsultarAlum(w_dni)* de la clase Alumno y a la operación *createAlumProf(Rp,Ra)* de la clase ProfAlum.

- Clase Pantalla BorrarProyecto
 - Operación *MostrarP_BorrarProyecto*. Esta operación muestra la pantalla de la función Borrar Proyecto.
- Clase BorrarProyecto Control
 - Operación que se ejecuta al hacer *new.BorrarProyectoControl*. Esta operación invoca a la operación *MostrarP_BorrarProyecto* de la clase Pantalla BorrarProyecto.
 - *BorrarProyecto(w_nombre)*. Esta operación invoca a la operación **Rp:=ConsultarProy(w_nombre)* de la clase Proyecto, a la operación *BorrarProy(Rp)* de la clase Proyecto, a la operación **Rap:=ConsultarAlumProy(w_nombre)* de la clase AlumProy y **BorrarAlumProy(Rap)* de la clase AlumProy.
- Clase Pantalla CambiarConfProy
 - Operación *MostrarP_CambiarConfProy*. Esta operación muestra la pantalla de la función Cambiar Nivel de Proyectos.
- Clase CambiarConfProy Control
 - Operación que se ejecuta al hacer *new.CambairConfProy*. Esta operación invoca a la operación *MostrarP_CambiarConfProy* de la clase Pantalla CambiarConfProy.
 - *CambiarConfProy(w_nmes, w_nsem, w_bmes, w_bsem)*. Esta operación invoca a la operación **Rp:=ObtenerP()* de la clase Sesión, a la operación *Rpc:=ObtenerPC(Rp)* de la clase ProfConfig y a la operación *CambiarConfProy(w_nmes, w_nsem, w_bmes, w_bsem, Rpc)* de la clase ProfConfig.
- Clase Pantalla CerrarSesion
 - Operación *MostrarP_CerrarSesion*. Esta operación muestra la pantalla de Cerrar Sesión.
- Clase CerrarSesion Control
 - Operación que se ejecuta al hacer *new.CerrarSesionControl*. Esta operación invoca a la operación *MostrarP_CerrarSesion* de la clase Pantalla CerrarSesion.
 - *CerrarSesion()*. Esta operación invoca a la operación **Rs:=ConsultarSesion()* de la clase Sesión y a la operación *Destruir()* de la clase Sesión.
- Clase Pantalla CrearCopia
 - Operación *MostrarP_CrearCopia*. Esta operación muestra la pantalla de Crear Copia de Seguridad.

- Clase CrearCopia Control

- Operación que se ejecuta al hacer *new.CrearCopiaControl*. Esta operación invoca a la operación *MostrarP_CrearCopia* de la clase Pantalla CrearCopia.
- *CrearCopia()*. Esta operación invoca a la operación **Rs:=ConsultarSesion()* de la clase Sesión, a la operación **Ra:=ConsultarAdmin(Rs)* de la clase Admin y a la operación *CrearCopia()* de la clase Admin.

- Clase Pantalla DescargarCSV

- Operación *MostrarP_DescargarSV*. Esta operación muestra la pantalla de Descargar CSV.

- Clase DescargarCSV Control

- Operación que se ejecuta al hacer *new.DescargarCSVControl*. Esta operación invoca a la operación *MostrarP_DescargarCSV* de la clase Pantalla DescargarCSV.
- *DescargarCSV(w_nombre)*. Esta operación invoca a la operación **Rp:=Consultar-Proy(w_nombre)* de la clase Proyecto y a la operación *DescargarCSV(Rp)* de la clase Stats.

- Clase Pantalla EstatProyecto

- Operación *MostrarP_EstatProyecto*. Esta operación muestra la pantalla de Estadísticas Proyecto.

- Clase EstatProyecto Control

- Operación que se ejecuta al hacer *new.EstatProyectoControl*. Esta operación invoca a la operación *MostrarP_EstatProyecto* de la clase Pantalla EstatProyecto.
- *EstatProyecto(w_nombre)*. Esta operación invoca a la operación **Rp:=Consultar-Proy(w_nombre)* de la clase Proyecto y a la operación *MostrarProy()* de la clase Proyecto.

- Clase Pantalla LoginProfesor

- Operación *MostrarP_LoginProfesor*. Esta operación muestra la pantalla de Login Profesor.

- Clase LoginProfesor Control

- Operación que se ejecuta al hacer *new.LoginProfesorControl*. Esta operación invoca a la operación *MostrarP_LoginProfesor* de la clase Pantalla LoginProfesor.
- *loginProfesor(w_login,w_pass)*. Esta operación invoca a la operación **Rp:=ConsultarPF(w_login)* de la clase Profesor y a la operación *createSess(Rp)* de la clase Sesión.

■ Clase Pantalla VincularProyecto

- Operación *MostrarP_VincularProyecto*. Esta operación muestra la pantalla de Vincular Proyecto.

■ Clase LoginProfesor Control

- Operación que se ejecuta al hacer *new.VincularProyectoControl*. Esta operación invoca a la operación *MostrarP_VincularProyecto* de la clase Pantalla VincularProyecto.
- *VincularProyecto(w_dni,w_nombre)*. Esta operación invoca a la operación **Ra:=ConsultarAlum(w_dni)* de la clase Alumno, a la operación **Rp:=ObtenerP(w_nombre)* de la clase Proyecto y a la operación *createAlumProy(Rp,Ra)* de la clase AlumProy.

■ Clase Proyecto

- Operación **Rp:=ConsultarProy(w_nombre)*. Esta operación devuelve un enlace al Proyecto con w_nombre.
- Operación *BorrarProy(Rp)*. Esta operación elimina el objeto Rp de la clase Proyecto.
- Operación *MostrarProy()*. Esta operación muestra por Pantalla las estadísticas del Proyecto.

■ Clase Stats

- Operación *CreateStats(Rp)*(constructor). Esta operación crea un objeto S de la clase Stats y actualiza los datos de la revisión concreta.
- Operación *DescargarCSV(Rp)*. Esta operación crea un archivo CSV de los datos del objeto Rp y permite su descarga.

■ Clase Alumno

- Operación *ConsultarAlum(w_dni)*. Esta operación comprueba si existe un alumno con w_dni.
- Operación *createAlum(w_nom,w_apl,w_dni,w_web)*(constructor). Esta operación crea un objeto de la clase Alumno A y le añade los datos w_nom, w_apl, w_dni, w_web. Además invoca a la operación *createAlumProf(Rp,Ra)*.

■ Clase Sesión

- Operación **Rp:=ObtenerP()*. Esta operación devuelve la relación con el objeto P que tiene iniciada la sesión.
- Operación **Rs:=ConsultarSesion()*. Esta operación devuelve la relación con el objeto S que tiene iniciada la sesión.
- Operación *createSess(Rp)*. Esta operación crea un objeto de la clase sesión con los datos de Rp (login y nombre).

- Operación *Destruir()*(destructor). Destruye la sesión actual.
- Clase ProfAlum
 - Operación *createAlumProf(Rp,Ra)*(constructor). Esta operación crea un objeto de la clase ProfAlum relacionado con Ra y Rp.
- Clase AlumProy
 - Operación **Rap:=ConsultarAlumProy(w_nombre)*. Esta operación comprueba devuelve una relación al objeto AlumProy cuyo nombre del proyecto es w_nombre.
 - Operación **BorrarAlumProy(Rap)*(destructor). Esta operación elimina los objetos Rap.
 - Operación *createAlumProy(Rp,Ra)*(constructor). Esta operación crea un objeto de la clase AlumProy relacionado con Rp y Ra.
- Clase ProfConfig
 - Operación **Rpc:=ObtenerPC(Rp)*. Esta operación devuelve un objeto de ProfConfig relacionado con Rp.
 - Operación *CambiarConfProy(w_nmes,w_nsem,w_bmes,w_bmes,Rpc)*. Esta operación modifica los datos del objeto Rpc.
- Clase Admin
 - Operación **Ra:=ConsultarAdmin(Rs)*. Esta operación devuelve una relación con el objeto de la clase Admin según la sesión actual.
 - Operación *CrearCopia()*. Esta operación crea un archivo con una copia de la base de datos del sistema.
- Clase Profesor
 - Operación **existe:=ConsultarPF(w_login,w_pass)*. Esta operación comprueba si existe un Profesor con esos datos.
 - Operación **Rp:=ConsultarProf(w_login,w_pass)*. Esta operación devuelve una relación al objeto de Profesor cuyo login y pass son w_login y w_pass.
- (...)

6.5. Capa de Modelos

6.5.1. Descripción

Como se dijo en el último apartado de introducción, la capa de Modelos o de Persistencia se define como la capa que se encarga del almacenamiento y la recuperación de los datos utilizando algún sistema de almacenamiento.

El sistema de almacenamiento más utilizado es una Base de Datos, pero también podría ser uno o un conjunto de archivos planos en formato XML. El tipo de base de datos no importa, lo que es importante es la API (*Interfaz de Programación de Aplicaciones*) suministrada por la implementación, ya que es ella la que determina cómo se accede a la información tras la puesta en marcha del sistema.

Normalmente, existe un Servidor de Base de Datos sobre el cual se carga la base de datos, para más tarde, poder acceder a él desde cualquier parte que interese. Los métodos más comunes para acceder a los datos son ODBC, JDBC, ADO y ADO.Net. Todos estos métodos utilizan el SQL como lenguaje para leer los datos de la base de datos.

Nuestro software tiene como sistema de almacenamiento de datos una Base de Datos controlada por el paquete MySQL. La base de datos de la aplicación está compuesta por todas las tablas que soportan la funcionalidad de la aplicación.

Abordaremos el estudio de la Base de Datos del sistema a través del Esquema Relacional. El uso de este esquema está muy extendido puesto que permite reflejar las entidades que conformarán el modelo y las relaciones existentes entre ellas.

Al igual que casi todos los sistemas de gestión de base de datos modernos, la información se almacena y trata utilizando el ***Modelo de Gestión de Bases de Datos Relacional***. En un sistema de base de datos relacional, los datos se organizan en Tablas. A continuación se procederá con el detalle de las tablas que se almacenarán en la base de datos del sistema.

6.5.2. Esquema Relacional

La información del sistema se almacena en tablas para poder ser tratado por el SGBD Relacional.

Así pues, el siguiente paso consiste en realizar la conversión del Modelo Conceptual de Datos o Modelo de Entidad-Relación (desarrollado en el capítulo anterior de este documento) a tablas y atributos. El resultado obtenido tras llevar a cabo este proceso es lo que se conoce como ***Esquema Relacional o Esquema de Base de Datos***.

Por lo tanto, el esquema ya normalizado es el siguiente:

- **Profesor** (login, Nombre, Apellidos, email, confirmado)
- **Alumno** (DNI, Nombre, Apellidos, web, email)
- **Proyecto** (Nombre, Web, Blog, SVN, blogXML, Rev, tipo, Prof)
- **Stats** (Rev, NombreProy, Desarrollador, Mensaje, Fecha, NLineasA, NLineasD, NAdd, NDel, NMod, AplA, AplD, AplM, DocA, DocD, WebA, WebD, WebM, ImgA, ImgD)
- **Usuario** (login, pass)
- **ProfConfig** (NombreProf, NMes, NSemana, BMes, BSemana, ANMes, ANSemana)
- **ProfAlum** (LOGprof, DNIalum)
- **AlumProy** (DNI, Nombre)
- **AlumDesarrollador** (DNI, Desarrollador)
- **Blog** (Nombre, Titulo, Fecha)
- **ProyFlags** (NombreProy, Actualizando)
- **Admin** (LOGIN, PASS, hora, min, dias)
- **ActFlags** (ActualTodo, Nombre, RevF, n, t, b)

Capítulo 7

Codificación

En la fase de codificación o implementación de un proyecto informático se construye una solución software utilizando las tecnologías y herramientas seleccionadas en la fase de diseño. Durante esta fase también se realizan tareas de integración, unión o conexión con componentes externos.

7.1. Tecnologías

Para el desarrollo de esta aplicación se han utilizado las tecnologías HTML, CSS, PHP (CodeIgniter), MySQL y JavaScript. A continuación, se presentan las diferentes tecnologías utilizadas en la implementación de la aplicación agrupadas por el nivel de diseño en el que se ha utilizado.

7.1.1. Nivel de Presentación (Vistas)

HTML

El HTML (*Hyper Text Markup Language*) o, en castellano, lenguaje de marcación de hipertexto, es el lenguaje de marcas de texto utilizado normalmente en World Wide Web (WWW). Fue creado por el físico nuclear Tim Berners-Lee en 1986, a partir de dos herramientas preexistentes: el concepto de hipertexto y el SGML. El concepto de hipertexto, también conocido como link o ancla, permite conectar dos elementos entre sí. Por otro lado SGML (*Standard Generalized Markup Language*) es un lenguaje estándar de marcación general que sirve para colocar etiquetas o marcas en un texto que indique cómo debe mostrarse.

HTML es una sintaxis que se utiliza para definir la estructura y ubicación de los elementos que se quieren mostrar en una página Web. Esta sintaxis también es capaz de definir relaciones entre componentes de un sitio Web gracias al uso de hipervínculos.

La interpretación de los documentos HTML suelen realizarla los navegadores Web, como por ejemplo: Netscape Navigator, Mozilla Firefox, Microsoft Internet Explorer o Google Chrome entre otros.

CSS

El termino CSS, acrónimo de *Cascading Style Sheets* (hojas de estilo en cascada), hace referencia al lenguaje formal usado para definir la presentación de documentos estructurados escritos en HTML, XML y XHTML por extensión. El World Wide Web Consortium (W3C) es el organismo encargado de formular la especificación de las hojas de estilo que servirán como estándar para los navegadores. La idea que se reside tras el desarrollo de CSS es separar la estructura de un documento de su presentación.

Ventajas derivadas de la utilización de CSS

- *Aumento de la legibilidad del documento HTML.* El documento es más fácil de leer y editar puesto que en él no se mezclan estructura y descripciones de estilo. Además, si no se utiliza la modalidad de CSS en línea, consistente en introducir la definición de estilo dentro del mismo documento de estructura, se reduce considerablemente el tamaño del documento.
- *Disminución del tiempo necesario para cambiar el aspecto global del sitio Web.* El control centralizado de la presentación de un sitio Web permite agilizar de forma considerable la actualización del mismo.
- *Aumento de la accesibilidad.* Los Navegadores permiten a los usuarios especificar y aplicar al sitio Web su propia hoja de estilo local, con lo que se aumenta considerablemente la accesibilidad. Por ejemplo, personas con deficiencias visuales pueden configurar su propia hoja de estilo para aumentar el tamaño del texto o remarcar más los enlaces.
- *Aumento de disponibilidad.* Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario. Por ejemplo, para ser impresa, mostrada en un dispositivo móvil o ser "leída" por un sintetizador de voz.

7.1.2. Nivel de Aplicación (Controladores y Modelos)

PHP

PHP es un acrónimo recursivo que, aunque ha tenido otros significados, actualmente significa PHP Hypertext Pre-processor. Fue creado originalmente por Rasmus Lerdorf en 1994,

sin embargo la implementación principal de PHP está producida por The PHP Group y sirve como estándar de facto de PHP. El lenguaje está publicado bajo la PHP License que es considerada por la Free Software Foundation como software libre.

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas Web dinámicas. Es un lenguaje de propósito general ampliamente utilizado que está diseñado especialmente para desarrollo Web. Generalmente se ejecuta en un servidor Web, el servidor toma el código en PHP como entrada y devuelve páginas Web como salida. Puede ser desplegado en la mayoría de los servidores Web y en casi todos los sistemas operativos de las diferentes plataformas sin coste alguno.

Su interpretación y ejecución es realizada por un servidor mientras que el cliente sólo recibe el resultado de la ejecución. Cuando el cliente hace una petición al servidor para que le envíe una página generada por un script PHP, el servidor ejecuta el intérprete de PHP. Este intérprete procesa el script para generar el contenido solicitado dinámicamente y se lo devuelve al servidor para que éste finalmente se lo sirva al cliente.

Ventajas derivadas de la utilización de PHP

- *Genera aplicaciones exportables a diferentes plataformas* por su definición como lenguaje multiplataforma.
- *Facilita la conexión a diferentes motores de Bases de Datos.* PHP dispone de capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- *Abundante documentación.* PHP dispone de una amplia documentación en su página Web oficial (<http://www.php.net/manual/es/>).
- *Fácil acceso.* Es un lenguaje de libre distribución que se presenta como una alternativa de fácil acceso para todo el mundo.
- *Tiene manejo de excepciones* (desde la versión PHP5).

Desventajas derivadas de la utilización de PHP

- Por su diseño *no puede ser compilado.*
- *Muy difícil de optimizar.*
- *Favorece la creación de código desordenado y complejo de mantener.*

CodeIgniter (Framework de PHP)

CodeIgniter es un conjunto de herramientas para personas que construyen su aplicación web usando PHP. Su objetivo es permitirle desarrollar proyectos mucho más rápido de lo que podría si lo escribiese desde cero, proveyendole un rico juego de librerías para tareas comunmente necesarias, así como una interface simple y estructura lógica para acceder a esas librerías. CodeIgniter le permite creativamente enfocarse en su proyecto minimizando la cantidad de código necesaria para una tarea dada.

Ventajas derivadas de la utilización de CodeIgniter

- *Es libre*: se encuentra bajo una licencia open source Apache/BSD-style.
- *Es liviano*: el núcleo del sistema sólo requiere unas pocas pequeñas librerías. Esto es en duro contraste a muchos entornos de trabajo que requieren significativamente más recursos. Las librerías adicionales son cargadas dinámicamente a pedido, basado en sus necesidades para un proceso dado, así que el sistema base es muy delgado y bastante rápido.
- *Es rápido y extensible*: el sistema puede ser fácilmente extendido a través del uso de plugins y librerías asistentes, o a través de extensión de clases o ganchos del sistema.

7.1.3. Nivel de Persistencia (Modelos)

MySQL

MySQL es un Sistema de Gestión de Bases de Datos Relacional, licenciado bajo la licencia GPL de GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca. Aunque MySQL es software libre, MySQL AB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de no ser así, se vulneraría la licencia GPL. Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Ventajas derivadas de la utilización de MySQL

- *Es posible aprovechar la potencia de sistemas multiprocesador, gracias a su implementación multihilo.*
- *Soporta gran cantidad de tipos de datos para las columnas.*
- *El sistema es multiplataforma por lo que dispone de APIs en gran cantidad de lenguajes (C, C++, Java, PHP, etc.).*
- *MySQL proporciona gran portabilidad entre sistemas.*
- *Soporta hasta 32 índices por tabla.*
- *Realiza una buena gestión de usuarios y passwords manteniendo muy buen nivel de seguridad en los datos.*

7.2. Herramientas

Todas las herramientas empleadas durante el desarrollo del proyecto, son herramientas gratuitas, de libre acceso que se encuentran disponibles en Internet. Todas ellas se pueden descargar desde sus páginas Web oficiales. Además, al ser de carácter libre son muy populares, por lo que muchas de ellas cuentan con una larga y extensa documentación generada por los mismos usuarios.

7.2.1. Apache

Durante el desarrollo de la aplicación EstDV se ha trabajado con la versión Apache 2.2. El servidor Apache es un servidor HTTP de código abierto para plataformas Unix, Windows, Macintosh, etc. que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPD 1.3, pero más tarde fue reescrito por completo. El servidor Apache se desarrolla dentro del proyecto HTTP Server (HTTPD) de la Apache Software Foundation. Ahora Apache presenta, entre otras características, mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido.

Apache tiene amplia aceptación en la red: desde 1996, Apache, es el servidor HTTP más usado. Alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70 % de los sitios Web en el mundo, sin embargo ha sufrido un descenso en su cuota de mercado en los últimos años.

Al ser EstDV una aplicación desarrollada en PHP, se ha utilizado el modulo PHP de Apache. La forma de funcionamiento es bastante sencilla de entender: cuando el servidor

recibe una petición de página con extensión *.php*, éste llama al intérprete de PHP, el cual procesa (interpreta) el código almacenado en la página solicitada y una vez procesado le devuelve el resultado al servidor Apache. Con toda la información ya procesada, el servidor Apache sólo tiene que servir el código HTML facilitado por el intérprete a los usuarios finales.

7.2.2. Gimp

El diseño de algunos gráficos usados en la parte visual de la aplicación se ha realizado gracias al programa de edición de imágenes GIMP. Esta herramienta se selecciono principalmente por ser software libre y estar disponible sin coste alguno. La versión de GIMP con la que se ha trabajado es GIMP 2.6.8.

GIMP (acrónimo recursivo de GNU *Image Manipulation Program*) es un programa de edición de imágenes, tanto de dibujos como de fotografías. Es un software englobado en el proyecto GNU y disponible bajo la licencia pública general de GNU.

La primera versión de GIMP se desarrolló para sistemas Unix y fue pensada especialmente para GNU/Linux. Existen versiones totalmente funcionales para Windows, Mac OS X, distribuciones Linux, etc. Se podría decir que es el programa de manipulación de gráficos disponible en más sistemas operativos pudiéndolo considerar como una de las alternativas más firmes a Adobe PhotoShop. Como datos curiosos a destacar decir que la biblioteca de controles gráficos GTK, desarrollada para GIMP, dio origen al entorno de escritorio de GNOME.

7.2.3. Gedit

Es un editor de textos compatible con UTF-8 para GNU/Linux, Mac OS X y Microsoft Windows. Diseñado como un editor de textos de propósito general, gedit enfatiza la simplicidad y facilidad de uso. Incluye herramientas para la edición de código fuente y textos estructurados, como lenguajes de marcado. Es el editor predeterminado de GNOME.

Esta herramienta ha sido de utilidad en el desarrollo de la aplicación como apoyo de la herramienta NetBeans. Además ha sido usada para la descripción de este documento debido a la facilidad de instalación de plugins (en concreto el plugin de latex, que es el lenguaje en el que está descrito este documento).

La aplicación está diseñada para funcionar sobre X Window System, usando las librerías GTK+ y GNOME. El editor está integrado con el resto de GNOME, soportando el arrastrado de archivos entre Nautilus hacia el cuerpo principal de la aplicación, uso del framework de impresión del gestor de GNOME y su sistema integrado de ayuda.

Algunas de las muchas características ofrecidas por Gedit son:

- Sintaxis coloreada y envoltura de sintaxis.
- Sintaxis coloreada a elección del usuario.
- Multi-Documento (pestañas).
- Soporte para Buscar/Reemplazar expresiones regulares.
- Soporte completo para arrastrar y colocar.
- Detención automática del estado del documento.
- Herramienta de Zoom.
- Funcionamiento bajo entornos multilingües.
- Resaltado de paréntesis y sangría.
- Grabación y reproducción de macros.
- Los lenguajes de programación admitidos son: C, C++, Java, C#, XML, HTML, PHP, JavaScript, makefile, Shell Script de UNIX, SQL, CSS, Pascal, Perl, lenguaje ensamblador, PostScript; entre muchos otros.

7.2.4. NetBeans

NetBeans es un entorno de desarrollo, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

NetBeans permite crear aplicaciones Web con PHP 5, un potente debugger integrado y además viene con soporte para Symfony un gran framework MVC escrito en php. Al tener también soporte para AJAX, cada vez más desarrolladores de aplicaciones LAMP o WAMP, están utilizando NetBeans como IDE.

7.2.5. Gantt Project

El diagrama de Gantt es una popular herramienta gráfica cuyo objetivo es mostrar el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado. A pesar de que, en principio, el diagrama de Gantt no indica las relaciones existentes entre actividades, la posición de cada tarea a lo largo del tiempo hace que se puedan identificar dichas relaciones e interdependencias. Fue Henry Laurence Gantt quien, entre 1910 y 1915, desarrolló y popularizó este tipo de diagrama en Occidente.

GanttProject es una herramienta gratuita de escritorio multiplataforma para la programación y gestión de proyectos. Se ejecuta en Windows, Linux y MacOSX, es libre y su código es opensource.

¿Qué puede hacer?: *Diagrama de Gantt* Crear una estructura de desglose de trabajo, dependencias dibujar, definir hitos.; *Los recursos* Asignar recursos humanos para trabajar en las tareas, ver su asignación en la tabla de carga de recursos.; *PERT* Generar diagrama PERT de diagrama de Gantt.; *La exportación* Guardar gráficos como imágenes PNG, generar informes PDF y HTML.; *Interoperar* Proyectos de importación y de exportación a los formatos de Microsoft Project. Exportación a hojas de cálculo con CSV.; *Colaborar* Compartir proyectos con sus colegas utilizando WebDAV..

7.2.6. Forja RedIris

RedIRIS es la red española para Interconexión de los Recursos Informáticos de las universidades y centros de investigación. Como tal provee de servicios de conexión a Internet a dichas instituciones. Fue fundada en el año 1988 como un proyecto del entonces Plan Nacional de I+D del Ministerio de Educación y Ciencia en colaboración con *Telefónica* a través de la fundación *Fundesco* y actualmente está gestionada por la Entidad Pública empresarial *Red.es* y financiada por el Plan Nacional de I+D+i.

Entre los servicios que ofrece está la Forja de Conocimiento Libre de la Comunidad RedIRIS, que ofrece a toda la comunidad un repositorio para desarrollar iniciativas libres en el entorno académico-científico universitario. Esta Forja se usa, por ejemplo, como forja oficial en el Concurso Universitario de Software Libre.

Además esta forja ha servido de alojamiento al desarrollo del proyecto. Este se encuentra en la siguiente dirección <https://forja.rediris.es/projects/estdv/>.

Capítulo 8

Pruebas

Las pruebas se han realizado como un proceso de ejecución de un programa con la intención de descubrir errores en el buen funcionamiento del sistema. El objetivo de estas pruebas son las de encontrar errores y subsanarlos. De entre todos los tipos de pruebas que existen, se comentarán los tipos de pruebas que se han usado en el desarrollo del sistema, además de los conjuntos de entrada para dichas pruebas.

8.1. Tipos de Pruebas

8.1.1. Pruebas de Validación y Verificación

Las pruebas de validación en la ingeniería de software son el proceso de revisión que el sistema de software producido cumple con las especificaciones y que cumple su cometido. Es normalmente una parte del proceso de pruebas de software de un proyecto, que también utiliza técnicas tales como evaluaciones, inspecciones, y tutoriales. La validación es el proceso de comprobar lo que se ha especificado es lo que el usuario realmente quería. Además se revisa si el resultado corresponde a la especificación del sistema, es decir, si se está construyendo el sistema de forma correcta.

8.1.2. Pruebas de Integración

Pruebas integrales o pruebas de integración son aquellas que se realizan en el ámbito del desarrollo de software una vez que se han aprobado las pruebas unitarias. Únicamente se refieren a la prueba o pruebas de todos los elementos unitarios que componen un proceso, hecha en conjunto, de una sola vez.

8.2. Conjuntos de Entrada

En el desarrollo del sistema se demarcan dos claros conjuntos de entrada distintos que, además, se dieron en momentos distintos en las etapas del desarrollo software.

8.2.1. Pruebas durante el Desarrollo

El primer conjunto de pruebas viene determinado por haber sido usado durante la etapa de desarrollo. Este conjunto se compone de valores aleatorios, inventados sin ningún tipo de patrón. Su finalidad fue probar el correcto uso de la base de datos por parte del sistema además de ver el correcto funcionamiento de la interfaz web.

8.2.2. Pruebas Finales

Para las pruebas finales el conjunto de pruebas fue el conjunto de proyectos del curso 2010/2011 presentados por los alumnos en la asignatura de Diseño de Videojuegos de Ingeniería Técnica Informática de Sistemas de Cádiz. Estos datos se caracterizan por ser datos reales, y extenderse a todas las ramas que monitoriza el sistema (SVN y blog).

Capítulo 9

Conclusiones

9.1. Valoración Personal

Desde el punto de vista académico, el desarrollo de la aplicación estDV ha sido un éxito. Ha cumplido todos los objetivos fijados al inicio del desarrollo. El cliente y usuario están contentos con él (en este caso el cliente es el usuario principal). Esta aplicación le permitirá mantener un control sobre los proyectos de sus alumnos, proyectos fin de carrera, o cualquier proyecto al que desee hacerle un seguimiento a través de un repositorio SVN. Esta aplicación supone una mejora en la gestión académica de los proyectos, por su comodidad y facilidad de acceso a la información.

Desde un punto de vista personal, este proyecto ha sido un gran éxito. Este proyecto me ha ayudado, además de a interiorizar todas las partes necesarias en el desarrollo de un proyecto, a aprender una rama de la informática que aun no conocía, el desarrollo web. Esto era un punto pendiente en mis conocimientos sobre programación que quería completar y el desarrollo de este proyecto fin de carrera me ha hecho completarlo con creces habiendo aprendido lenguajes como CSS, HTML, PHP o CodeIgniter.

Todo este trabajo de aprendizaje, programación, errores, más errores ha sido el que ha hecho que obtenga una visión distinta en el desarrollo de proyectos, y una forma de trabajo más organizada y eficaz.

Además este proyecto me ha permitido haberme convertido en cada uno de los integrantes de un equipo que trabajaría en una aplicación como esta. Como son por ejemplo: ingeniero de requisitos, desarrollador, diseñador gráfico, entre otros. Me ha hecho ver el mismo proyecto desde distintos puntos de vista y aprender mucho sobre cada uno de estos papeles.

9.2. Futuras Ampliaciones

La aplicación cumple los objetivos indicados al inicio del desarrollo, de hecho, tiene algunas funcionalidades incorporadas, siempre buscando la comodidad de su uso.

A pesar de esto, siempre quedan cosas que se podrían hacer para aumentar la calidad de una aplicación. En este caso, bajo mi punta de vista hay algunas cosas que estDV podría incorporar y no han sido incorporadas por, mayoritariamente, falta de tiempo:

- Sistema: Incorporar servicio de correo electrónico, para comunicar a usuarios y administrador eventos (nuevas cuentas, proyectos abandonados, etc).
- Administrador: Incorporar la opción de restaurar una copia de seguridad.
- Usuario: Incorporar opciones para modificar la apariencia de la aplicación, para mayor comodidad visual.
- Web: Añadir la posibilidad de leer tweets además de blogs.
- Web: Leer repositorios Git además de Svn.

Anexos A

Manual de Usuario

En esta sección vamos a explicar todas las funcionalidades del sistema disponibles a cada tipo de usuario. Vamos a explicarlas paso a paso y mostrando las pantallas por las que tendrá que pasar para llegar a completar la funcionalidad con éxito. Aunque el sistema está preparado para que sea intuitivo y fácil de asimilar, aquí veremos una descripción detallada de cada acción.

Primero, al acceder al sitio web tendremos que loguearnos en el sistema. Para loguearnos podremos hacerlo como **Alumno**, **Profesor** o **Administrador**.

A.1. Sección para Alumnos

A.1.1. Login Alumno

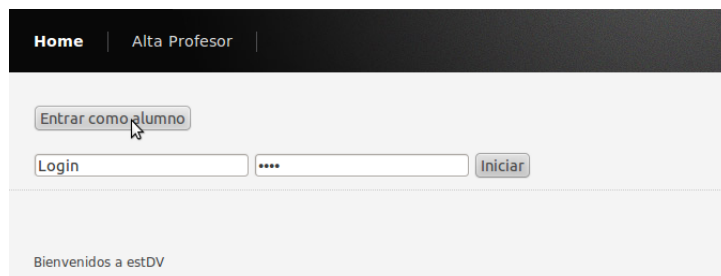


Figura A.1: Pantalla Inicial

Al entrar en el sistema nos encontramos con esta pantalla, por defecto el sistema espera un login de Profesor. Sin embargo podemos loguearnos también como Alumnos. Para ello pulsamos sobre el boton marcado **Entrar como alumno**.

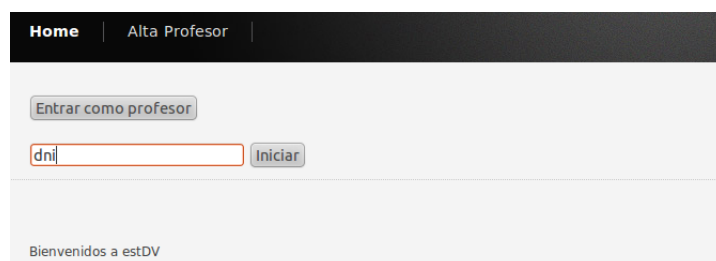


Figura A.2: Pantalla Login Alumno

Ahora estamos en la pantalla para loguearnos como Alumno. Para loguearnos tendremos que insertar nuestro DNI y pulsar el botón **Iniciar**.

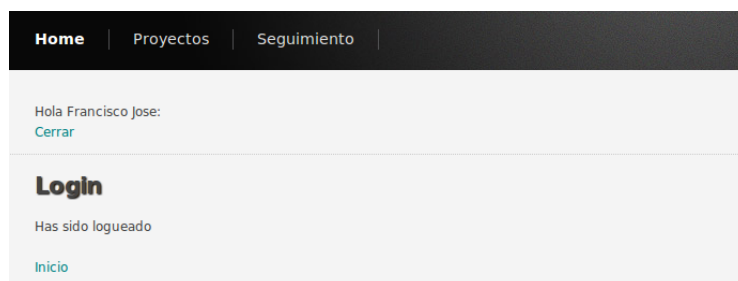


Figura A.3: Pantalla Éxito Login Alumno

Una vez logueados accedemos a la zona de Alumnos. Aquí podremos acceder a **Proyectos** o **Seguimiento**.

A.1.2. Mostrar Estadísticas según Proyecto

Un vez que el Alumno se ha logueado en el sistema, podrá acceder a la página de estadísticas referentes a sí mismo y a uno de los Proyectos en los que esté registrado.

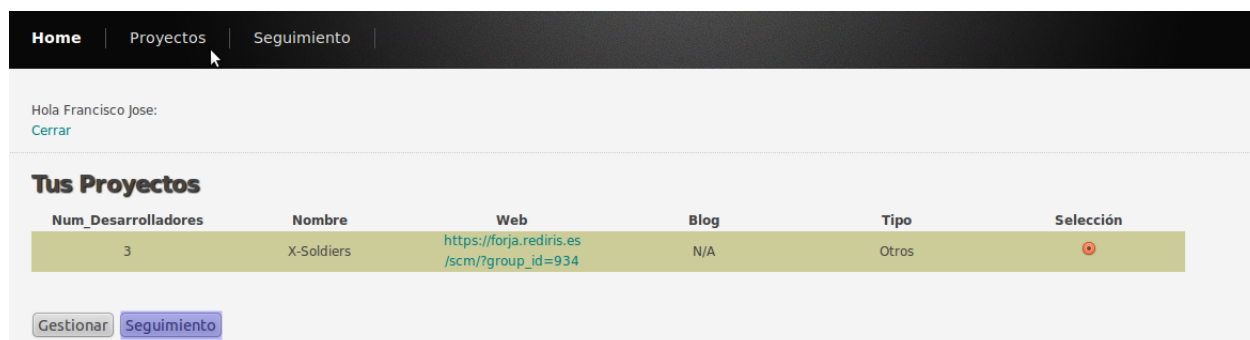


Figura A.4: Pantalla Lista Proyectos

Tras seleccionar la opción **Proyectos** el sistema mostrará una lista de Proyectos y tendremos dos opciones: **Gestionar** y **Seguimiento**. Seleccionamos la opción **Seguimiento** y accederemos a las estadísticas del Alumno dentro del Proyecto seleccionado.

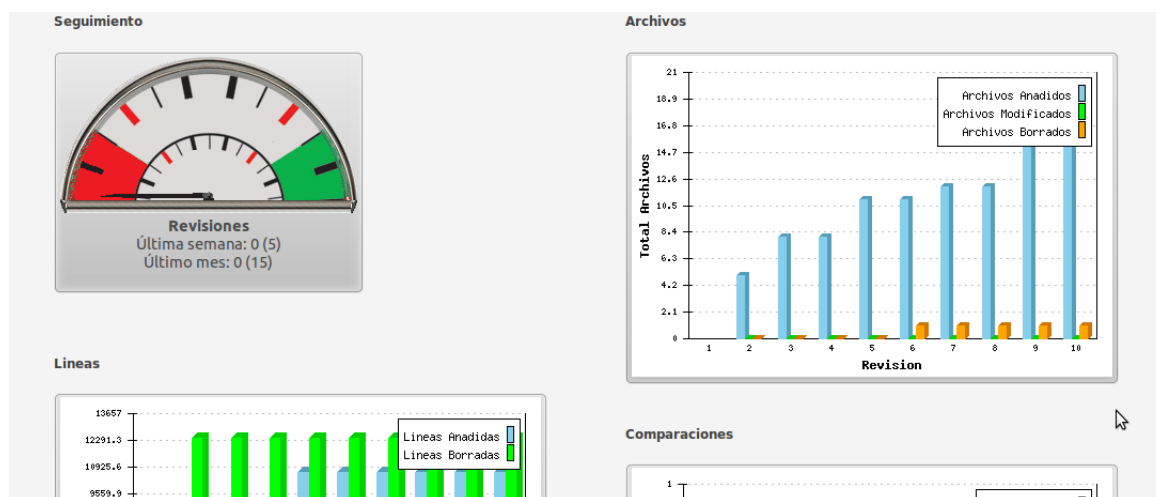


Figura A.5: Pantalla Estadísticas de Alumno según Proyecto

A.1.3. Mostrar Estadísticas según Profesor

Un vez que el Alumno se ha logueado en el sistema, podrá acceder a la página de estadísticas referentes a sí mismo y a uno de los Profesores a los que esté asociado.

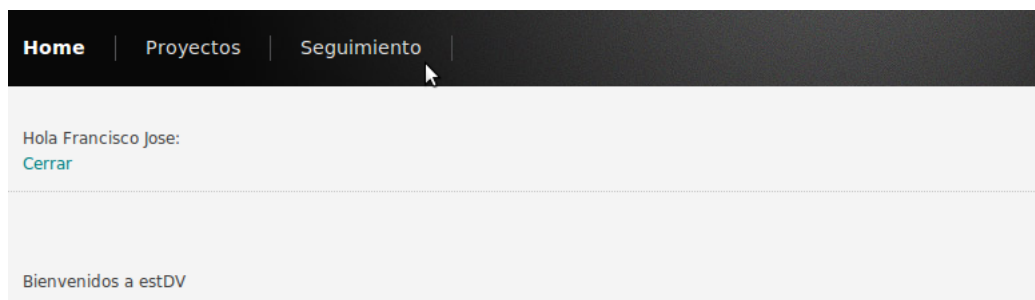


Figura A.6: Pantalla selección de opción

Tras seleccionar la opción de **Seguimiento** marcada en la imagen, entraremos en las estadísticas del Alumno centradas en un Profesor. Si solo está asociado a un Profesor (será lo normal) accederá a las estadísticas directamente. En caso de tener más de un profesor deberá seleccionar el que desee de una lista.

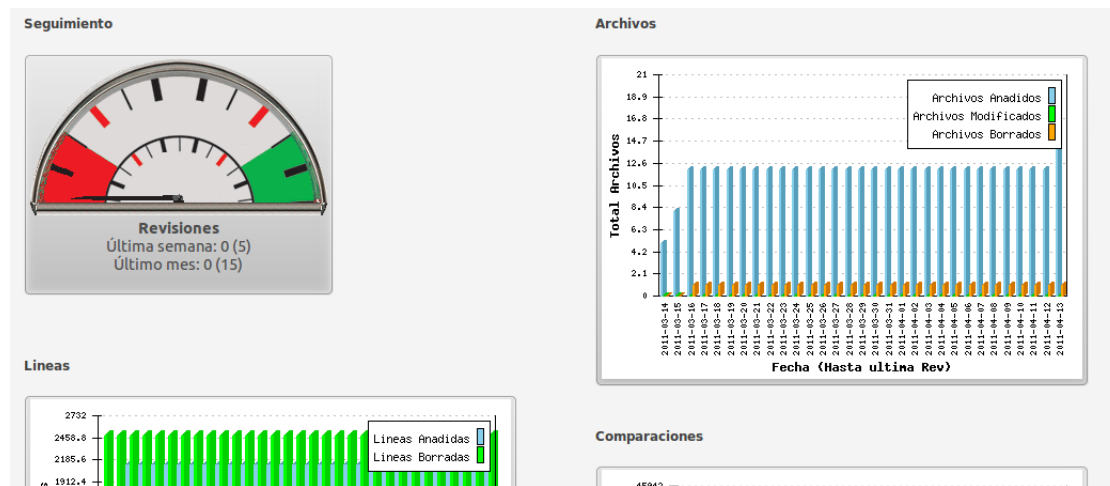
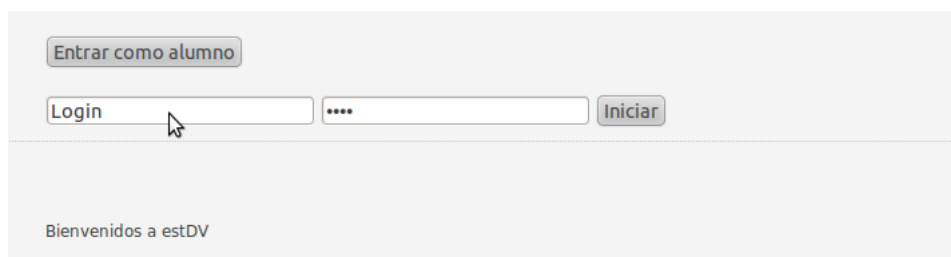


Figura A.7: Pantalla Estadísticas de Alumno según Profesor

A.2. Sección para Profesores

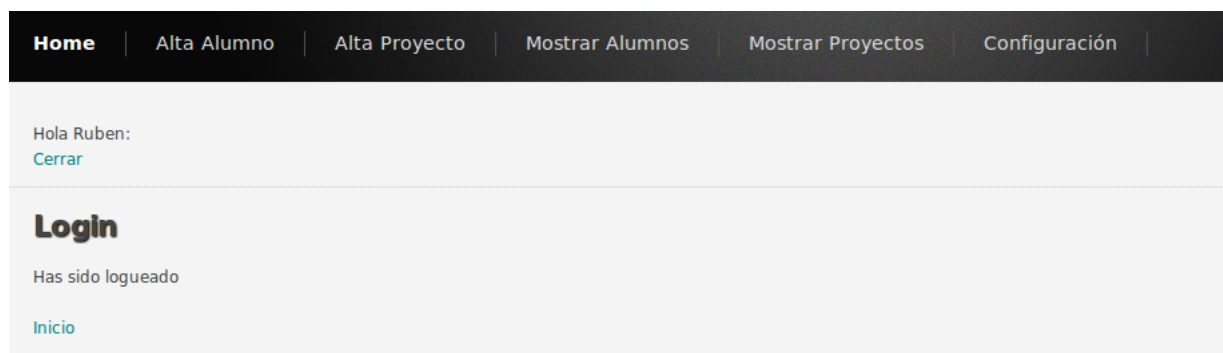
A.2.1. Login Profesor



The screenshot shows a login interface with a light gray background. At the top, there is a button labeled "Entrar como alumno". Below it, there are two input fields: the first is labeled "Login" and the second contains four dots, representing a password field. A mouse cursor is pointing at the "Login" field. To the right of the password field is a button labeled "Iniciar". At the bottom of the form area, the text "Bienvenidos a estDV" is displayed.

Figura A.8: Pantalla Inicial

Al entrar en el sistema nos encontramos con esta pantalla, por defecto el sistema espera un login de Profesor. Para entrar introducimos el **login** y **pass** y pulsamos el botón **Iniciar**. En caso de que sean correctos entraremos en el sistema.



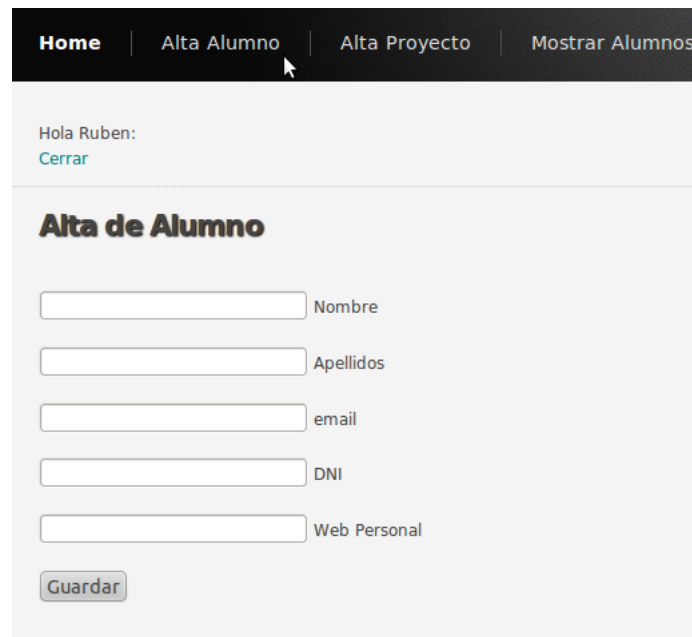
The screenshot shows the user interface after a successful login. At the top, there is a dark navigation bar with the following menu items: "Home", "Alta Alumno", "Alta Proyecto", "Mostrar Alumnos", "Mostrar Proyectos", and "Configuración". Below the navigation bar, the user is greeted with "Hola Ruben:" followed by a "Cerrar" link. The main content area has a heading "Login" and a message "Has sido logueado" with an "Inicio" link below it.

Figura A.9: Pantalla Login Profesor

Una vez logueados accedemos a la zona de Profesores. Aquí podremos acceder a **Alta Alumno**, **Alta Proyecto**, **Mostrar Alumnos**, **Mostrar Proyectos** o **Configuración**.

A.2.2. Alta Alumno

Para que un Profesor de a un Alumno de alta deberá, tras haberse logueado con éxito, seleccionar la opción **Alta Alumno**.



The screenshot shows a web application interface. At the top, there is a dark navigation bar with four links: 'Home', 'Alta Alumno' (which is highlighted with a mouse cursor), 'Alta Proyecto', and 'Mostrar Alumnos'. Below the navigation bar, there is a light gray area with the text 'Hola Ruben:' and a blue link 'Cerrar'. The main content area is titled 'Alta de Alumno' and contains five input fields with labels: 'Nombre', 'Apellidos', 'email', 'DNI', and 'Web Personal'. Below these fields is a button labeled 'Guardar'.

Figura A.10: Pantalla Alta Alumno

En esta pantalla deberemos introducir los datos del Alumno (**Nombre, Apellidos, email, DNI, Web**. La Web se puede dejar en blanco). Una vez relleno le damos al botón **Guardar**. Si todo está bien pasaremos a una pantalla donde se muestran los datos concretos del Alumno que se ha creado.

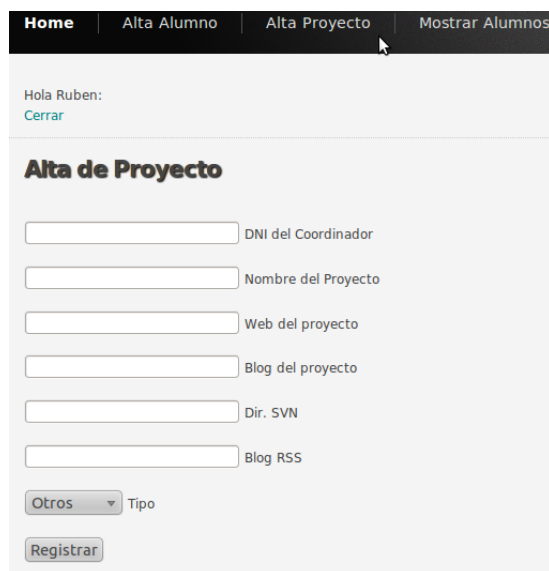


The screenshot shows a web application interface. At the top, there is a dark navigation bar with five links: 'Home', 'Alta Alumno', 'Alta Proyecto', 'Mostrar Alumnos', and 'Mostrar Proyectos'. Below the navigation bar, there is a light gray area with the title 'Confirmación alta usuario.'. Under the title, there is text showing 'Nombre: Pepa' and 'DNI: 48968302Z'. At the bottom of this area is a blue link 'Inicio'.

Figura A.11: Pantalla Alumno Creado

A.2.3. Alta Proyecto

Para que un Profesor de a un Proyecto de alta deberá, tras haberse logueado con éxito, seleccionar la opción **Alta Proyecto**.



The screenshot shows a web application interface with a dark navigation bar at the top containing links: Home, Alta Alumno, Alta Proyecto (highlighted with a mouse cursor), and Mostrar Alumnos. Below the navigation bar, a user greeting 'Hola Ruben:' is displayed with a 'Cerrar' link. The main content area is titled 'Alta de Proyecto' and contains a form with the following fields: 'DNI del Coordinador', 'Nombre del Proyecto', 'Web del proyecto', 'Blog del proyecto', 'Dir. SVN', and 'Blog RSS'. Below these fields is a dropdown menu labeled 'Otros' with a downward arrow, followed by the text 'Tipo'. At the bottom of the form is a 'Registrar' button.

Figura A.12: Pantalla Alta Proyecto

En esta pantalla deberemos introducir los datos del Proyecto (**DNI del Coordinador, Nombre, Web, Blog, SVN, Blog RSS, Tipo**. La Web, Blog y Blog RSS se pueden dejar en blanco). Una vez relleno le damos al botón **Guardar**. Si todo está bien pasaremos a una pantalla donde se muestran los datos concretos del Proyecto que se ha creado.



The screenshot shows the 'Confirmación alta Proyecto' screen. It features the same dark navigation bar as the previous screen. Below the navigation bar, the user greeting 'Hola Ruben:' is shown with a 'Cerrar' link. The main content area is titled 'Confirmación alta Proyecto' and displays the following information: 'Nombre: Marano', 'Web:', 'blog:', and 'Inicio' (a link).

Figura A.13: Pantalla Proyecto Creado

A.2.4. Listar Alumnos

Para listar los Alumnos asociados al Profesor logueado tendremos que seleccionar la opción **Mostrar Alumnos**.



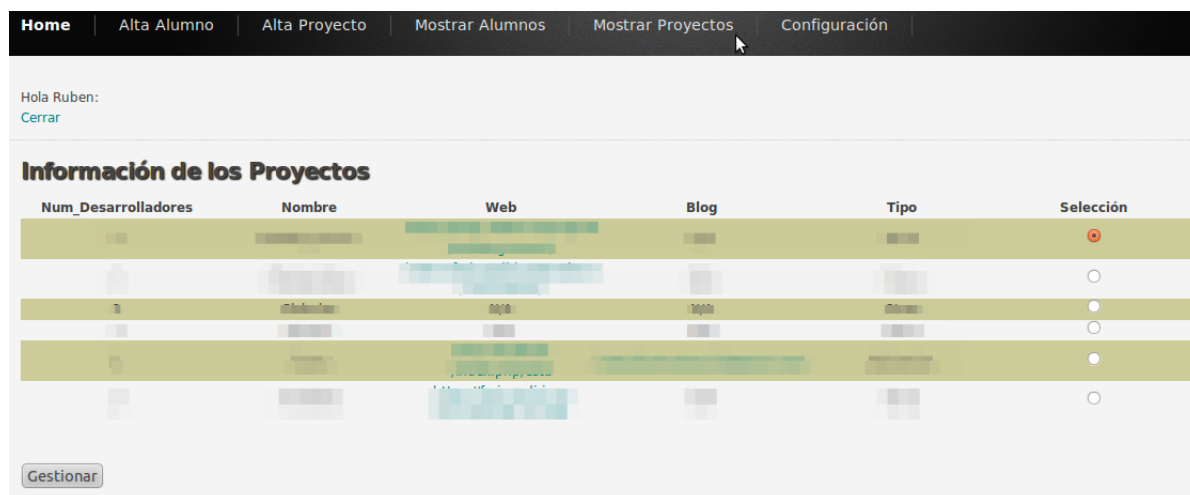
Figura A.14: Pantalla Lista Alumnos

En esta pantalla encontramos la lista de Alumnos que el Profesor ha dado de alta con una serie de opciones: **Seguimiento**, **Seguimiento por Proyecto**, **Mostrar Proyectos**, **Vincular Proyecto**, **Vincular Desarrollador**, **Borrar**.

- La opción **Seguimiento** se corresponde a la opción **Mostrar estadísticas del Alumno según Profesor**.
- La opción **Seguimiento por Proyecto** se corresponde a la opción **Mostrar estadísticas del Alumno según un Proyecto**.
- La opción **Mostrar Proyectos** muestra una lista de Proyectos asociados al Alumno.

A.2.5. Listar Proyectos

Para listar los Proyectos asociados al Profesor logueado tendremos que seleccionar la opción **Mostrar Proyectos**.



Num_Desarrolladores	Nombre	Web	Blog	Tipo	Selección
					<input checked="" type="radio"/>
					<input type="radio"/>
1					<input type="radio"/>
					<input type="radio"/>
					<input type="radio"/>

Gestionar

Figura A.15: Pantalla Lista Proyectos

En esta pantalla encontramos la lista de Proyectos que el Profesor ha dado de alta con la opción **Gestionar**.

A.2.6. Configurar Nivel Proyectos

Para configurar el nivel (revisiones y entradas al blog) de un Proyecto vamos a sección de **Configuración**.



Figura A.16: Pantalla de Configuración

Por defecto entramos en la configuración de Proyectos, así que introduciremos los datos (**Revs por semana y mes, Entradas al blog por semana y mes**) y pulsamos el botón **Guardar**.

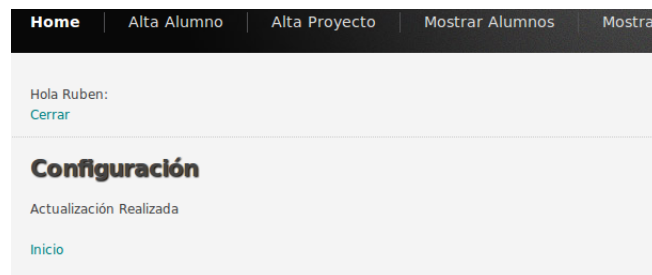


Figura A.17: Pantalla Configuración Actualizada

A.2.7. Configurar Nivel Alumnos

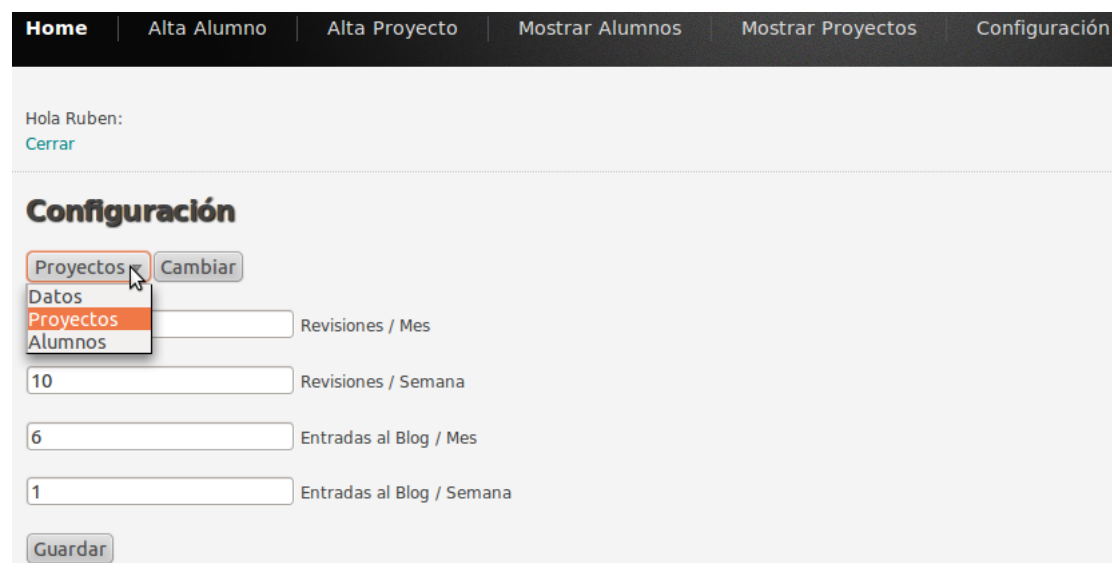
Para configurar el nivel (revisiones y entradas al blog) de un Alumnos vamos a sección de **Configuración**.



The screenshot shows the 'Configuración' page with a navigation bar at the top containing links: Home, Alta Alumno, Alta Proyecto, Mostrar Alumnos, Mostrar Proyectos, and Configuración. Below the navigation bar, a user greeting 'Hola Ruben:' is followed by a 'Cerrar' link. The main section is titled 'Configuración' and contains a dropdown menu set to 'Proyectos' with a 'Cambiar' button next to it. Below this are four input fields: 'Revisiones / Mes' (value 36), 'Revisiones / Semana' (value 10), 'Entradas al Blog / Mes' (value 6), and 'Entradas al Blog / Semana' (value 1). A 'Guardar' button is at the bottom.

Figura A.18: Pantalla de Configuración

Por defecto entramos en la configuración de Proyectos.



This screenshot shows the 'Configuración' page with the 'Proyectos' dropdown menu open. The menu options are 'Datos', 'Proyectos' (highlighted), and 'Alumnos'. The 'Cambiar' button is still visible next to the dropdown. The input fields for 'Revisiones / Mes' (36), 'Revisiones / Semana' (10), 'Entradas al Blog / Mes' (6), and 'Entradas al Blog / Semana' (1) are visible below the dropdown, along with the 'Guardar' button at the bottom.

Figura A.19: Pantalla Cambiar Tipo de Configuración



Figura A.20: Pantalla Configuración de Alumnos

Como queremos ir a la configuración de Alumnos desplegamos el menú señalado para cambiar a la configuración de Alumnos y pulsamos el botón **Cambiar**.

Ahora debemos introducir los datos que se nos piden (**Revs por semana y mes**) y pulsamos el botón **Guardar**.

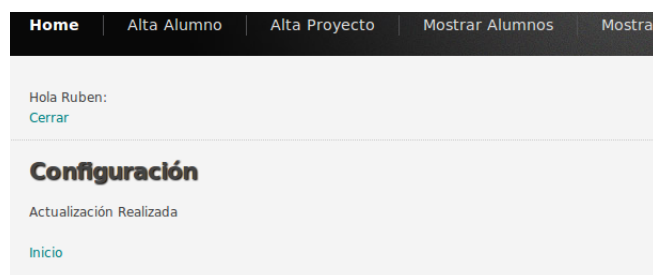
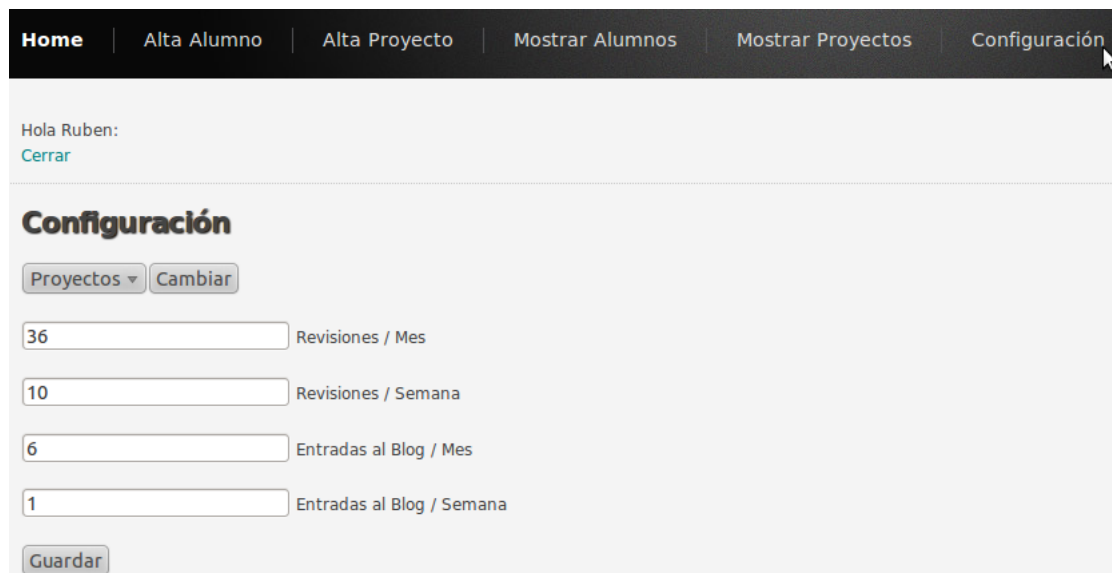


Figura A.21: Pantalla Configuración Actualizada

A.2.8. Cambiar Contraseña

Para cambiar la contraseña con la que accede el sistema el Profesor que está logueado accedemos a la sección de **Configuración**.



Home | Alta Alumno | Alta Proyecto | Mostrar Alumnos | Mostrar Proyectos | Configuración

Hola Ruben:
[Cerrar](#)

Configuración

Proyectos ▼ Cambiar

36 Revisiones / Mes

10 Revisiones / Semana

6 Entradas al Blog / Mes

1 Entradas al Blog / Semana

Guardar

Figura A.22: Pantalla de Configuración

Por defecto entramos en la configuración de Proyectos.



Home | Alta Alumno | Alta Proyecto | Mostrar Alumnos | Mostrar Proyectos | Configuración

Hola Ruben:
[Cerrar](#)

Configuración

Proyectos ▼ Cambiar

- Datos
- Proyectos**
- Alumnos

Revisiones / Mes

10 Revisiones / Semana

6 Entradas al Blog / Mes

1 Entradas al Blog / Semana

Guardar

Figura A.23: Pantalla Cambiar Tipo de Configuración



Figura A.24: Pantalla Configurar Contraseña

Como queremos ir a la configuración de contraseña desplegamos el menú señalado para cambiar al apartado **Datos** y pulsamos el botón **Cambiar**.

Ahora debemos introducir los datos que se nos piden (**antigua contraseña y nueva contraseña** (dos veces)) y pulsamos el botón **Guardar**.

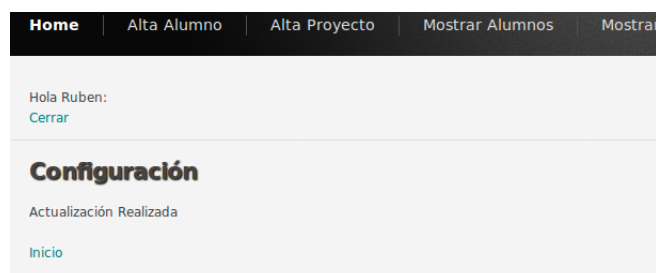


Figura A.25: Pantalla Contraseña Actualizada

A.2.9. Vincular Proyecto

Dentro del listado de Alumnos, una de las opciones es vincular un Proyecto al Alumno. Para esto vamos a la **Lista de Alumnos** y seleccionamos un Alumno.

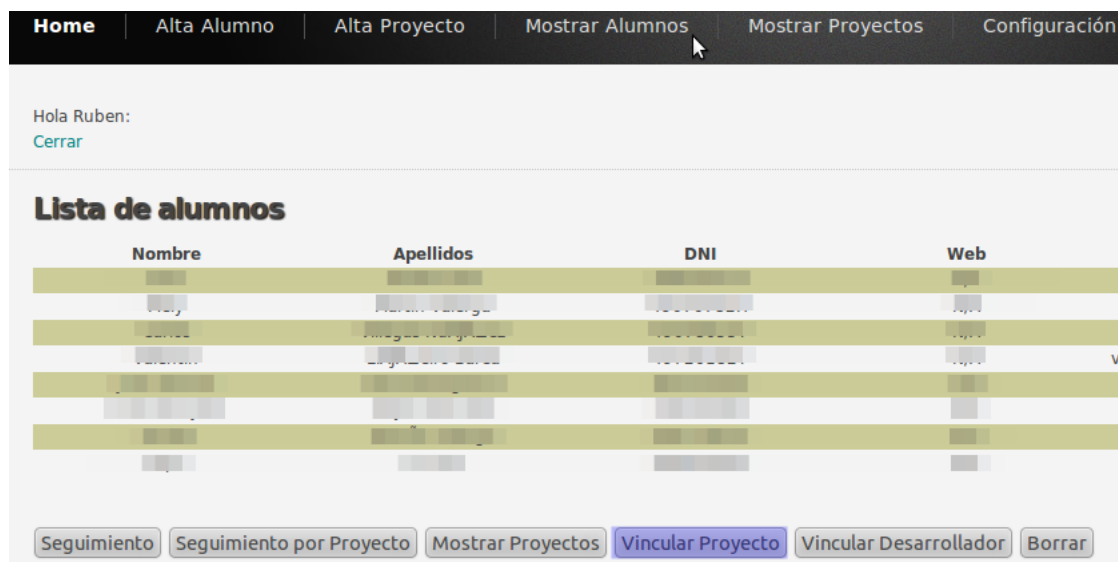


Figura A.26: Pantalla Lista Alumnos

Una vez seleccionado el Alumno, pulsamos el botón **Vincular Proyecto**. Nos llevará a otra pantalla con la lista de Proyectos del Profesor.



Figura A.27: Pantalla Lista Proyectos

Aquí seleccionamos el Proyecto con el que deseamos vincular al Alumno y pulsamos el botón **Vincular**. Con este completamos la asociación entre el Alumno y el Proyecto.

A.2.10. Vincular Desarrollador

Dentro del listado de Alumnos, una de las opciones es vincular un Desarrollador al Alumno. Para esto vamos a la **Lista de Alumnos** y seleccionamos un Alumno.

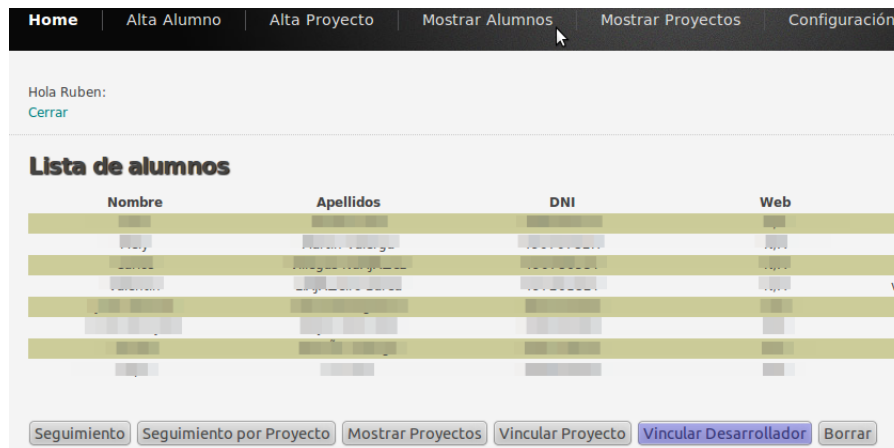


Figura A.28: Pantalla Lista Alumnos

Una vez seleccionado el Alumno, pulsamos el botón **Vincular Desarrollador**. Nos llevará a otra pantalla con la lista de Desarrolladores de los Proyectos a los que el Alumno está asociado.



Figura A.29: Pantalla Lista Desarrolladores

Aquí seleccionamos el Desarrollador con el que deseamos vincular al Alumno y pulsamos el botón **Vincular**. Con este completamos la asociación entre el Alumno y el Desarrollador.

A.2.11. Borrar Alumno

Dentro del listado de Alumnos, una de las opciones es **Borrar Alumno** al Alumno. Para esto vamos a la **Lista de Alumnos** y seleccionamos un Alumno.

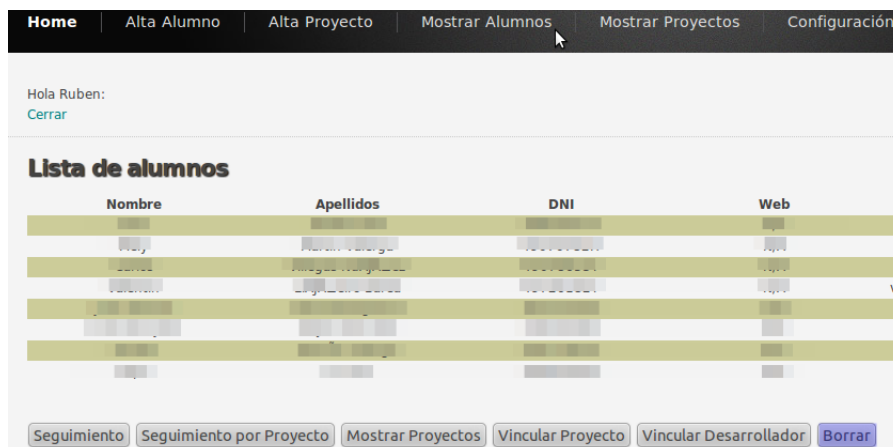


Figura A.30: Pantalla Lista Alumnos

Una vez seleccionado el Alumno, pulsamos el botón **Borrar Alumno**. Nos llevará a otra pantalla donde nos preguntará si realmente deseamos borrar al Alumno.



Figura A.31: Pantalla Confirmar

Aceptamos la eliminación del Alumno y nos llevará a una pantalla donde nos dirá que el Alumno ha sido borrado con éxito.

A.2.12. Borrar Proyecto

Dentro del listado de Proyecto, tenemos la opción de **Gestionar** el Proyecto, donde podremos realizar algunas acciones sobre el Proyecto que hayamos seleccionado. Para esto vamos a la **Lista de Proyecto**, seleccionamos un Proyecto y pulsamos la opción **Gestionar**.

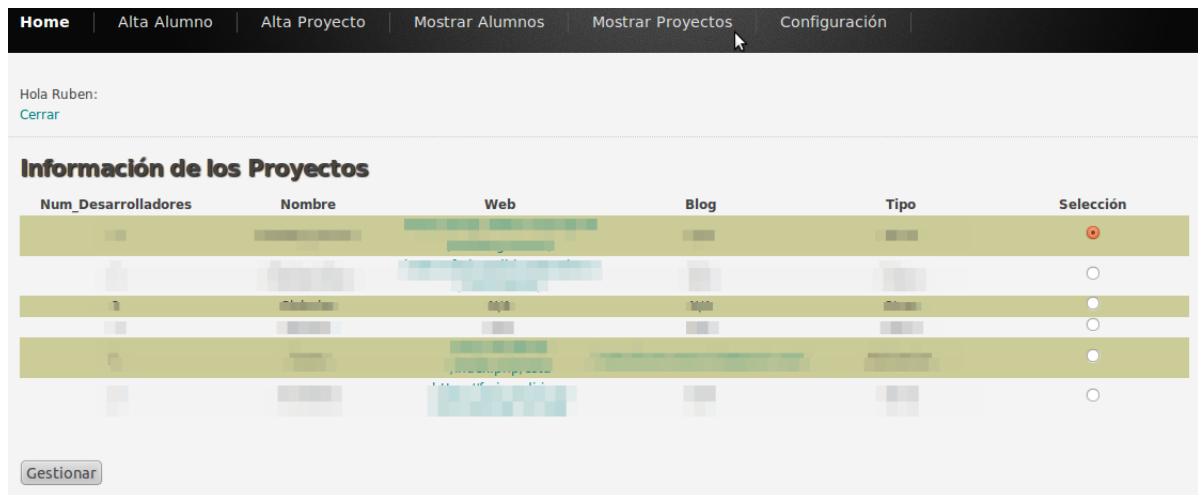


Figura A.32: Pantalla Lista Proyectos

Una vez entramos en la pantalla de gestión nos encontramos información sobre el Proyecto y algunas opciones (**Modificar**, **Borrar**, **Estadísticas**, **CSV**). Para borrar el proyecto seleccionamos la opción **Borrar**.



Figura A.33: Pantalla Gestionar



Figura A.34: Pantalla Confirmar Borrar proyecto

Tras seleccionar la eliminación del Proyecto nos pedirá una confirmación para ver si estamos seguros de borrarlo.

Aceptamos la confirmación y se nos notificará que el Proyecto ha sido borrado con éxito.



Figura A.35: Pantalla Éxito Borrar proyecto

A.2.13. Mostrar Estadísticas (Proy)

Dentro del listado de Proyecto, tenemos la opción de **Gestionar** el Proyecto, donde podremos realizar algunas acciones sobre el Proyecto que hayamos seleccionado. Para esto vamos a la **Lista de Proyecto**, seleccionamos un Proyecto y pulsamos la opción **Gestionar**.

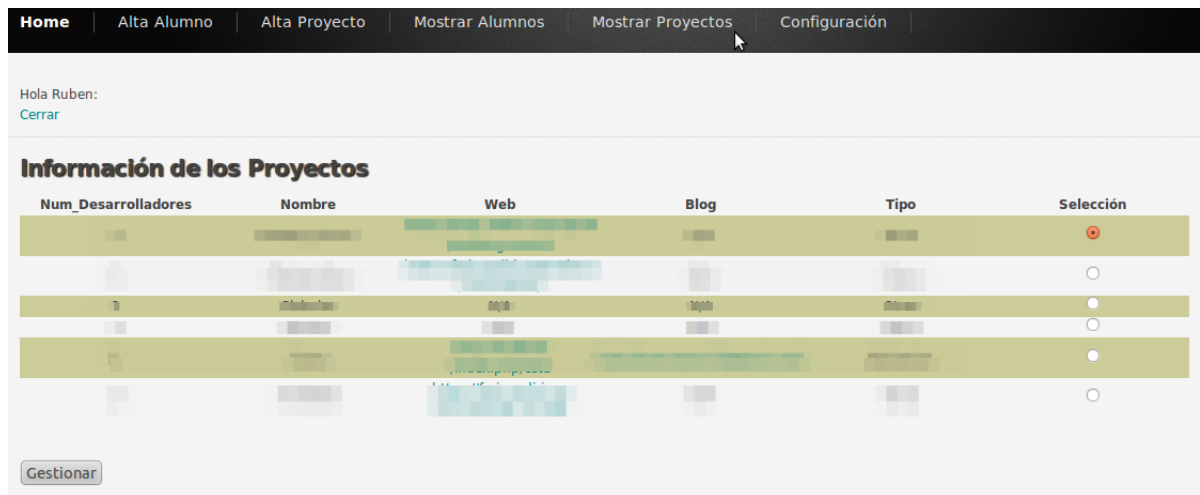


Figura A.36: Pantalla Lista Proyectos

Una vez entramos en la pantalla de gestión nos encontramos información sobre el Proyecto y algunas opciones (**Modificar**, **Borrar**, **Estadísticas**, **CSV**). Para borrar el proyecto seleccionamos la opción **Estadísticas**.

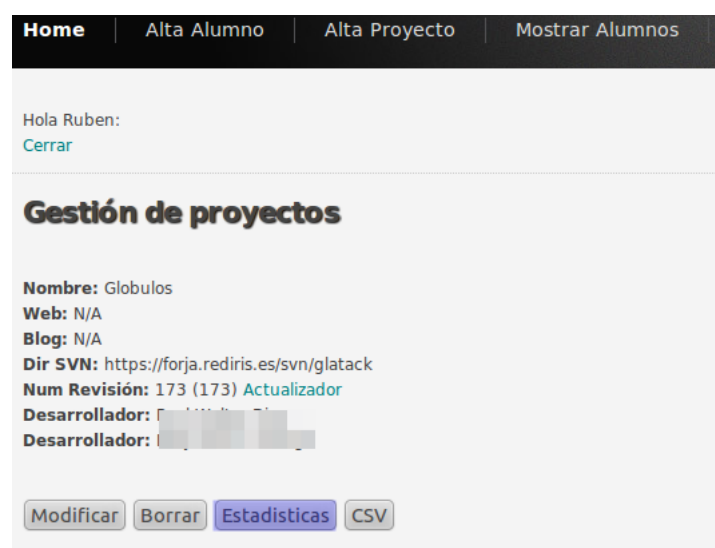


Figura A.37: Pantalla Gestionar

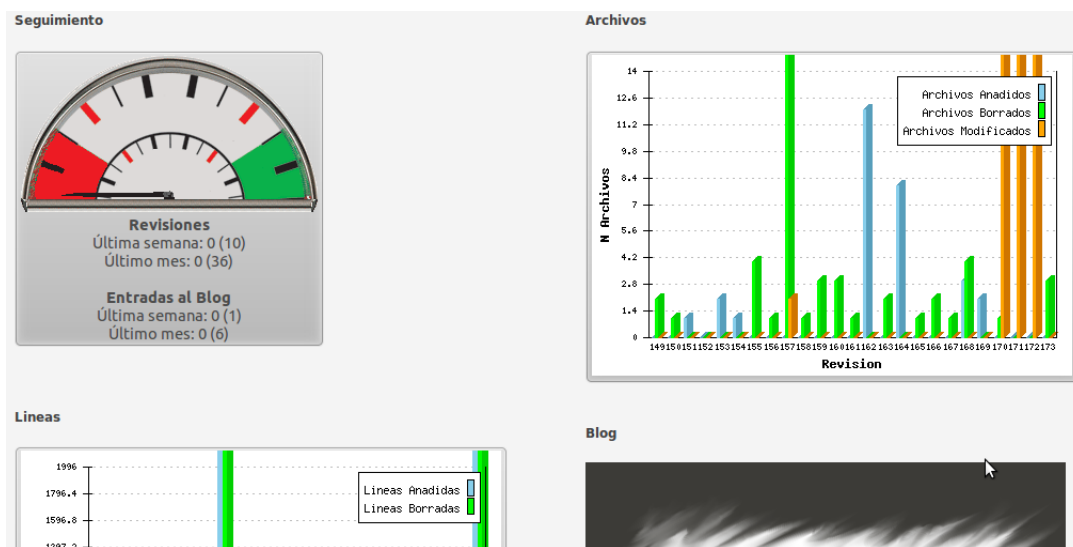


Figura A.38: Pantalla Mostrar Estadísticas de Proyecto

En esta pantalla tendremos las estadísticas centradas en el Proyecto que hayamos seleccionado.

A.2.14. Actualizar Proyecto

Dentro del listado de Proyecto, tenemos la opción de **Gestionar** el Proyecto, donde podremos realizar algunas acciones sobre el Proyecto que hayamos seleccionado. Para esto vamos a la **Lista de Proyecto**, seleccionamos un Proyecto y pulsamos la opción **Gestionar**.

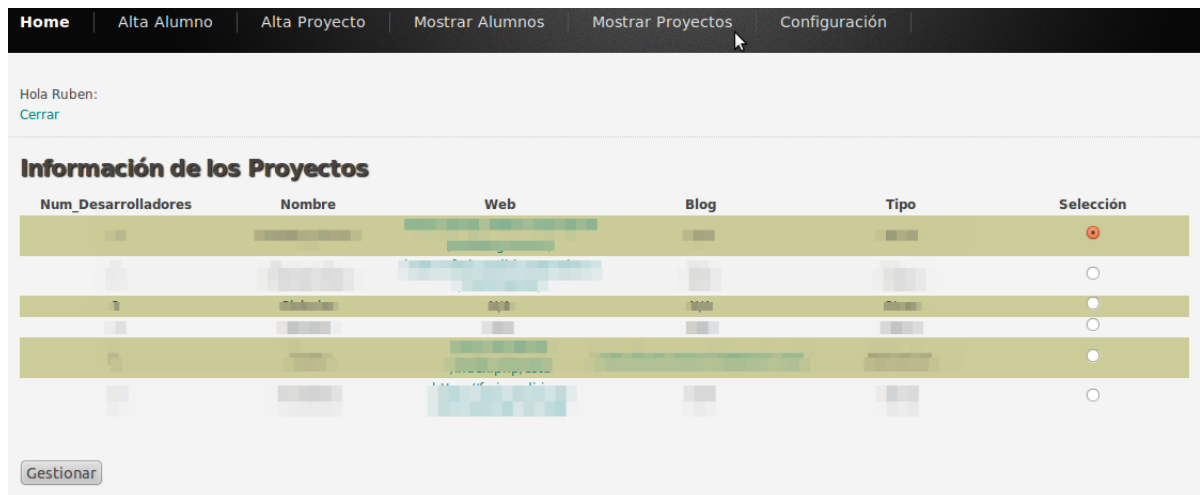


Figura A.39: Pantalla Lista Proyectos

Una vez entramos en la pantalla de gestión nos encontramos información sobre el Proyecto y algunas opciones (**Modificar**, **Borrar**, **Estadísticas**, **CSV**). Para borrar el proyecto seleccionamos la opción **Actualizador**.

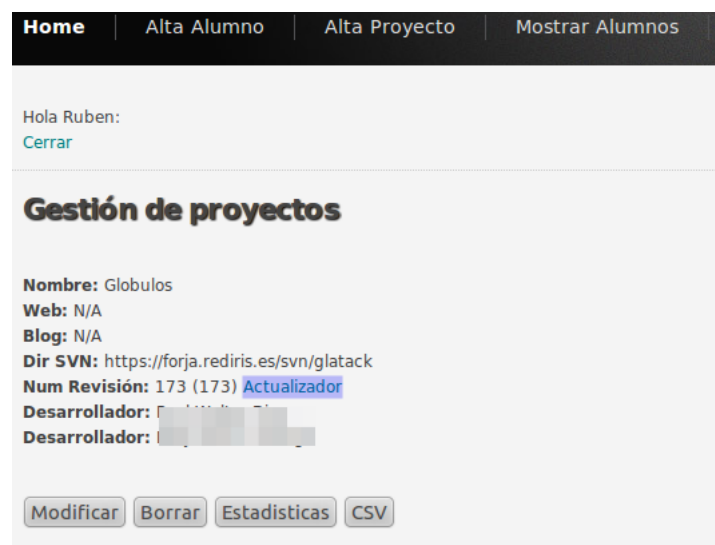


Figura A.40: Pantalla Gestionar

En esta pantalla tendremos la opción de poner a descargar la información del Proyecto en caso de que no esté descargando ya. Además podremos ver una monitorización de la descarga.

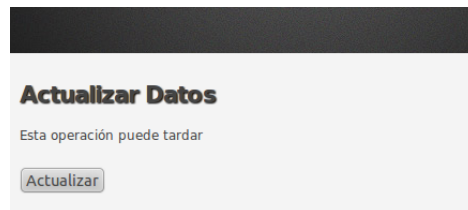


Figura A.41: Pantalla Actualizar Proyecto

A.2.15. Descargar CSV

Dentro del listado de Proyecto, tenemos la opción de **Gestionar** el Proyecto, donde podremos realizar algunas acciones sobre el Proyecto que hayamos seleccionado. Para esto vamos a la **Lista de Proyecto**, seleccionamos un Proyecto y pulsamos la opción **Gestionar**.

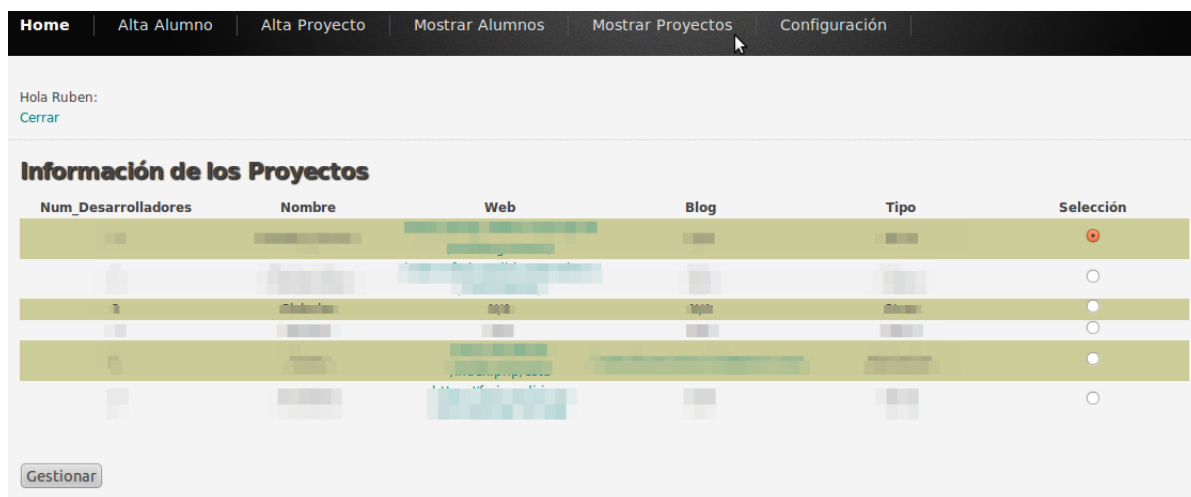


Figura A.42: Pantalla Lista Proyectos

Una vez entramos en la pantalla de gestión nos encontramos información sobre el Proyecto y algunas opciones (**Modificar, Borrar, Estadísticas, CSV**). Para borrar el proyecto seleccionamos la opción **CSV**.



Figura A.43: Pantalla Gestionar

Nos saltará la opción de descargar el archivo a nuestro ordenador.

A.2.16. Alta Profesor

Para darse de alta como Profesor en el sistema será necesario no tener ninguna sesión iniciada en el mismo. Es decir, estaremos en la pantalla inicial.

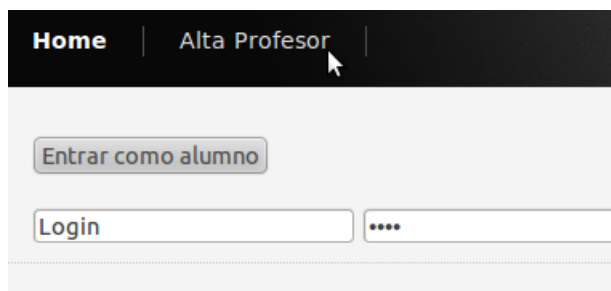


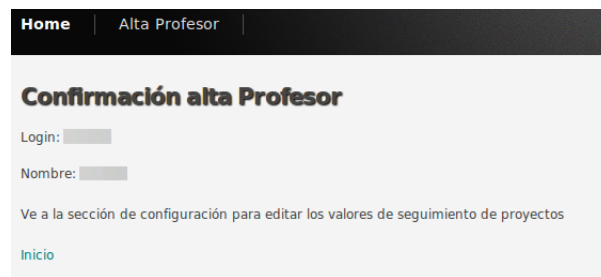
Figura A.44: Pantalla Inicio

Accederemos a una pantalla donde debemos introducir los datos de la nueva cuenta (**Nombre, Apellidos, Contraseña x2, email, Login**).



Figura A.45: Pantalla Gestionar

En caso de que todos los datos estén correctos se mostrará una pantalla donde se confirma los datos de la cuenta.



The screenshot shows a web application interface with a dark navigation bar at the top containing the links 'Home' and 'Alta Profesor'. The main content area has a light gray background and is titled 'Confirmación alta Profesor'. Below the title, there are two input fields: 'Login:' and 'Nombre:'. Below these fields, there is a line of text: 'Ve a la sección de configuración para editar los valores de seguimiento de proyectos'. At the bottom left of the content area, there is a blue link labeled 'Inicio'.

Figura A.46: Pantalla Gestionar

Para que la cuenta esté totalmente operativa, un administrador deberá de aceptarla como válida en el sistema.

A.3. Sección para Administrador

A.3.1. Login Administrador

Al entrar en el sistema nos encontramos con esta pantalla, por defecto el sistema espera un login de Profesor. Para entrar como administrador pulsamos el botón en la parte baja de la pantalla **Administración**.

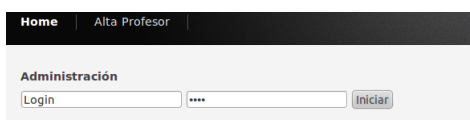


Figura A.47: Pantalla Login Administrador

Para loguearnos introducimos el login y la pass de administrador. Una vez logueados accedemos a la zona de Administración. Aquí podremos acceder a **Profesores, Alumnos, Proyectos, Nuevas Cuentas, Actualizar, Copia de Seguridad o Configuración**.

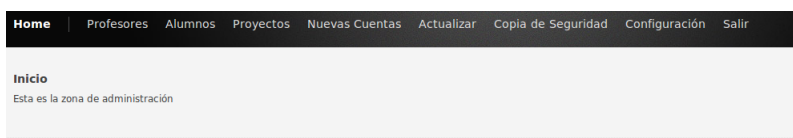


Figura A.48: Pantalla Login Administrador

A.3.2. Listar Profesores

Para listar los Profesores que existen en el sistema tendremos que seleccionar la opción **Profesores** tras habernos logueado como Administrador.

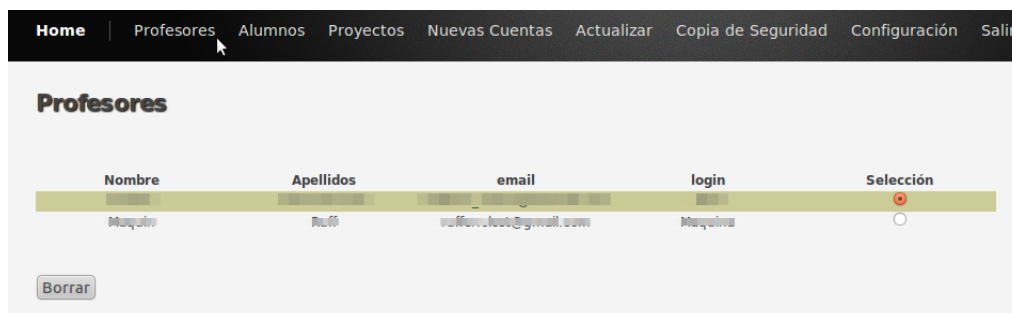
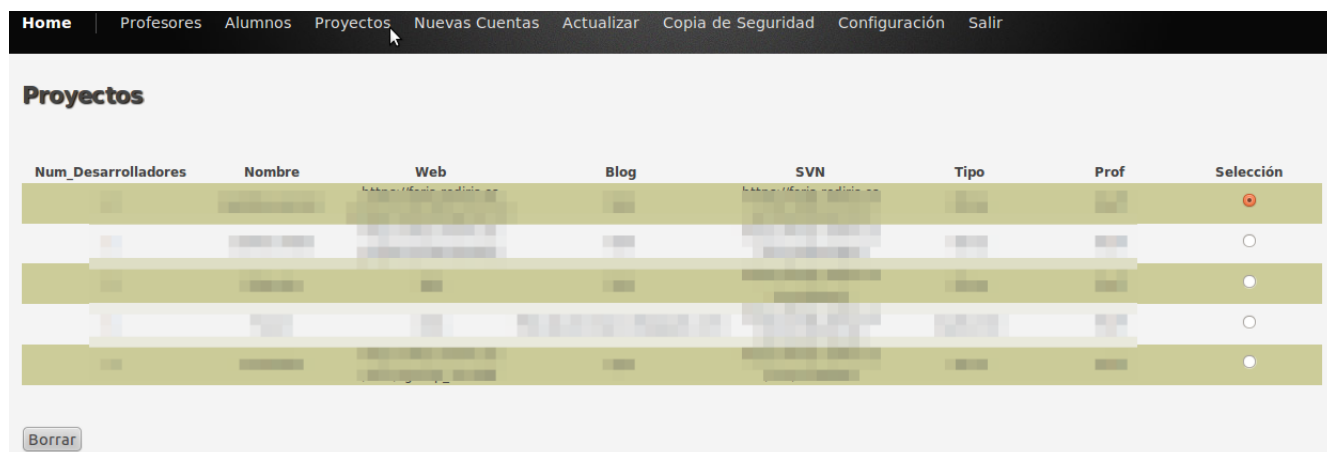


Figura A.49: Pantalla Lista Profesores

En esta pantalla tenemos la opción de eliminar el Profesor que seleccionemos y todo lo que este haya creado a través del botón **Borrar**.

A.3.3. Listar Todos los Proyectos

Para listar los Proyectos que existen en el sistema tendremos que seleccionar la opción **Proyectos** tras habernos logueado como Administrador.



Num_Desarrolladores	Nombre	Web	Blog	SVN	Tipo	Prof	Selección
							<input checked="" type="radio"/>
							<input type="radio"/>
							<input type="radio"/>
							<input type="radio"/>
							<input type="radio"/>

Borrar

Figura A.50: Pantalla Lista Proyectos

En esta pantalla tenemos la opción de eliminar el Proyecto del sistema a través del botón **Borrar**.

A.3.4. Listar Todos los Alumnos

Para listar los Alumnos que existen en el sistema tendremos que seleccionar la opción **Alumnos** tras habernos logueado como Administrador.



Figura A.51: Pantalla Lista Alumnos

En esta pantalla tenemos la opción de eliminar el Alumnos del sistema a través del botón **Borrar**.

A.3.5. Listar Nuevas Cuentas

Para listar las nuevas cuentas que existen en el sistema sin haber sido rechazadas o aceptadas tendremos que seleccionar la opción **Nuevas Cuentas** tras habernos logueado como Administrador.

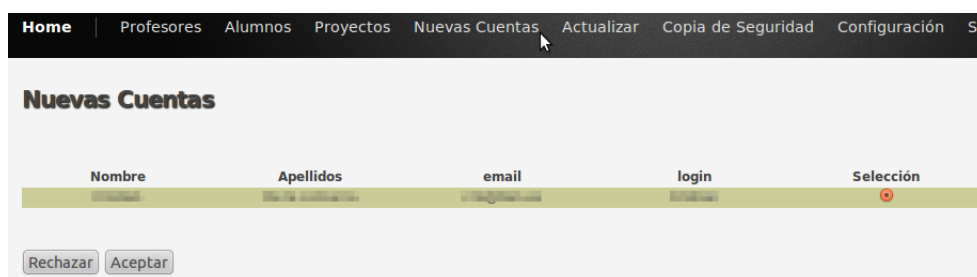


Figura A.52: Pantalla Lista de Nuevas Cuentas

En esta pantalla tenemos la opción de aceptar o rechazar las nuevas cuentas de Profesor a través de los botones **Aceptar** y **Rechazar**.

A.3.6. Lista de Copias de Seguridad

Para listar la copias de seguridad de la base de datos que existen en el sistema tendremos que seleccionar la opción **Copia de Seguridad** tras habernos logueado como Administrador.

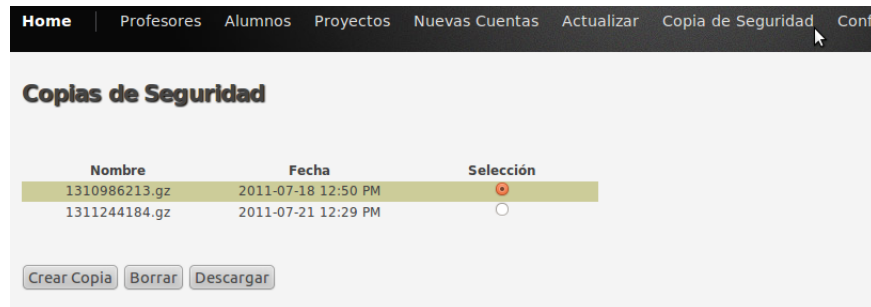


Figura A.53: Pantalla Lista de Copias de Seguridad

En esta pantalla tres acciones posibles:

- **Crear Copia:** Crea una nueva copia de la base de datos.
- **Borrar:** Elimina del sistema una copia de seguridad seleccionada.
- **Descargar:** Da la posibilidad de descargar a tu ordenador la copia de seguridad de la base de datos seleccionada.

A.3.7. Actualizar Todo

Para actualizar todos los Proyectos que existen en el sistema tendremos que seleccionar la opción **Actualizar** tras habernos logueado como Administrador.



Figura A.54: Pantalla Actualizar Todo

Para empezar la actualización pulsamos el botón **Actualizar Todo**.



Figura A.55: Pantalla Actualizar Todo

Confirmamos que queremos realizar la actualización con el botón **Actualizar**. Se actualizarán del sistema los proyectos uno a uno y podremos ver una monitorización de la actualización.



Figura A.56: Pantalla Actualizar Todo

A.3.8. Cambiar Contraseña de Administrador

Para cambiar la contraseña con la que accede el sistema el Administrador que está logueado accedemos a la sección de **Configuración**.

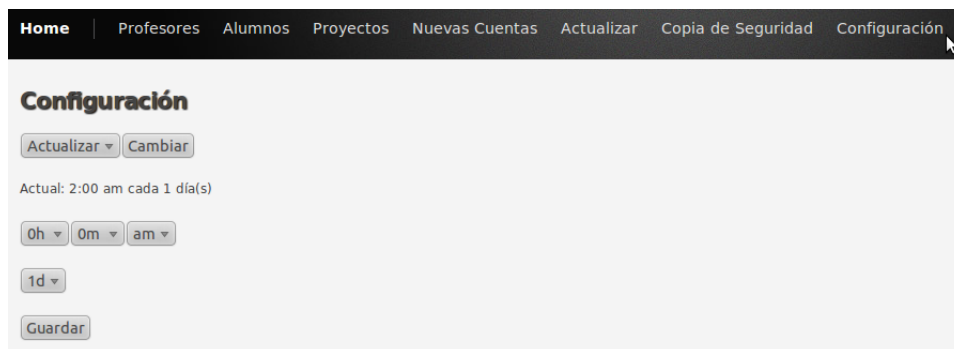


Figura A.57: Pantalla de Configuración

Por defecto entramos en la configuración de Actualización.



Figura A.58: Pantalla Cambiar Tipo de Configuración

Como queremos ir a la configuración de contraseña desplegamos el menú señalado para cambiar al apartado **Datos** y pulsamos el botón **Cambiar**.



The screenshot shows a web application interface with a dark navigation bar at the top containing links: Home, Profesores, Alumnos, Proyectos, Nuevas Cuentas, Actualizar, Copia de Seguridad, and Cont. Below the navigation bar, the main content area is titled 'Configuración'. Under this title, there is a 'Datos' dropdown menu and a 'Cambiar' button. Below these, there are three input fields: the first is labeled 'Contraseña' and contains six asterisks; the second and third are both labeled 'Nueva contraseña'. At the bottom of the form is a 'Guardar' button.

Figura A.59: Pantalla Configurar Contraseña

Ahora debemos introducir los datos que se nos piden (**antigua contraseña y nueva contraseña** (dos veces)) y pulsamos el botón **Guardar**.



Figura A.60: Pantalla Contraseña Actualizada

A.3.9. Cambiar Configuración de actualización

Para cambiar la contraseña con la que accede el sistema el Administrador que está logueado accedemos a la sección de **Configuración**.



Figura A.61: Pantalla de Configuración

Por defecto entramos en la configuración de Actualización. Introducimos los datos que se nos piden (**Hora, Minutos, Días**) y pulsamos el botón **Guardar**.



Figura A.62: Pantalla Contraseña Actualizada

A.4. Cerrar Sesión

A.4.1. Cerrar Sesión de Alumno

Para cerrar la sesión de un Alumno pulsaremos el botón **Cerrar** bajo el menú del sistema.

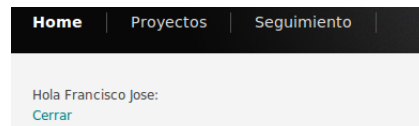


Figura A.63: Pantalla de Cerrar Sesión

A.4.2. Cerrar Sesión de Profesor

Para cerrar la sesión de un Profesor pulsaremos el botón **Cerrar** bajo el menú del sistema.

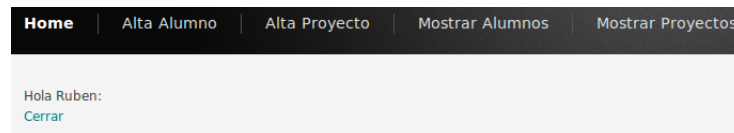


Figura A.64: Pantalla de Cerrar Sesión

A.4.3. Cerrar Sesión de Administrador

Para cerrar la sesión de un Administrador pulsaremos el botón **Salir** al final del menú del sistema.

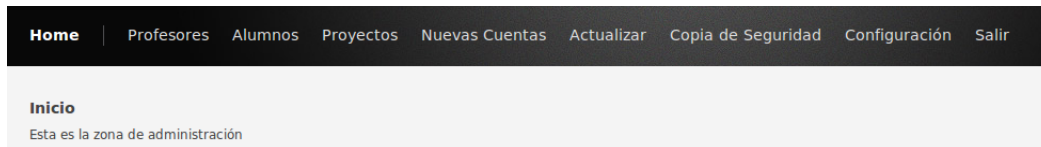


Figura A.65: Pantalla de Cerrar Sesión

Anexos B

Manual de Instalación

B.1. Introducción

En este capítulo explicaremos como instalar la aplicación. EstDV puede ser instalado en un servidor local o en un servidor remoto contratado a una empresa de alojamiento especializada. Este manual presenta los pasos necesarios para la instalación en un servidor local, pero la instalación en el servidor remoto es bastante sencilla, ya que la mayor diferencia entre los procesos reside en subir los archivos que forman la aplicación al servidor.

A continuación, y para facilitar aún más la instalación de la aplicación en un servidor local, presentaremos las herramientas necesarias, como obtenerlas e instalarlas. Cabe remarcar que la instalación está orientada en un sistema Ubuntu, ya que es el sistema en el que se ha desarrollado el sistema.

B.2. El Servidor

Los programas que necesitaremos para la instalación de la aplicación serán los siguientes:

- Servidor de páginas Web.
- Intérprete de PHP.
- Servidor de Bases de Datos.

En este apartado vamos a hablar sobre estas aplicaciones y explicar como instalarlas. Las aplicaciones que usaremos se caracterizan por ser programas *open source*, es decir, de tecnología abierta. Esto es así por varias razones. Principalmente, porque el software open source es gratis, lo que reduce los gastos de instalación, ya que no tenemos que pagar por ellos. Por otra parte los programas de tecnología abierta son tan eficientes y seguros como los programas de pago y, en muchos casos, más.

B.2.1. Instalación del Servidor Apache

El programa que vamos a emplear como servidor de páginas web es el Apache (<http://www.apache.org>). Es el servidor web más extendido en Internet. La mayoría de los sitios web que visitamos a diario están corriendo sobre este servidor. Las razones son varias. Por una parte, es gratuito. No tenemos que pagar nada para emplearlo. Por otra parte, es uno de los servidores más robustos que existen, muy seguro ante los ataques en la Red.

Para su instalación en Ubuntu bastará con:

- Abrir shell.
- Escribir: *sudo apt-get install apache2*.

B.2.2. Instalación del intérprete de PHP

La aplicación objeto de este proyecto funciona mediante el uso de un intérprete. Cuando el usuario solicita una página desarrollada mediante tecnología, el intérprete de PHP (<http://www.php.net>), instalado en el equipo servidor, lee el código de dicha página y lo ejecuta. A continuación genera una página HTML como resultado de la ejecución, y se la entrega al servidor web, quien se la envía al cliente. PHP es el lenguaje de scripts interpretados. En ese sentido el concepto es similar a JavaScript, dejando de lado el hecho de que este último se interpreta y ejecuta en el lado del cliente, y PHP lo hace en el lado del servidor. Por lo demás, la filosofía de trabajo es similar, aunque un lenguaje de servidor siempre ofrece más recursos y posibilidades que uno de cliente.

Para su instalación en Ubuntu bastará con:

- Abrir shell.
- Escribir: *sudo apt-get install php5*.
- Escribir: *sudo apt-get install php5-dev*.

B.2.3. Instalación de la Base de Datos MySQL

En el mercado existen diversos sistemas de bases de datos. Nosotros hemos elegido para la aplicación MySQL, por varias razones. Por una parte, es gratuito (aunque también existen licencias comerciales). Por otra parte, los datos se almacenan en archivos que tienen un tamaño mucho menor que otras bases de datos. Además, el motor de datos de MySQL es mucho más rápido, tanto grabando datos como localizándolos y recuperándolos, que el de otras bases de datos. Eso sin contar con que MySQL ofrece una gran seguridad sobre la integridad de los datos almacenados.

Para su instalación en Ubuntu bastará con:

- Abrir shell.
- Escribir: *sudo apt-get install mysql-server*. Se te pedirá crear la contraseña del administrador cuando se instale el programa (En versiones antiguas no se crea con la instalación. Debemos crearla manualmente, nada más instalar el servidor. Para hacerlo escribiremos en el terminal: *sudo /usr/bin/mysqladmin -u root password clavenueva*).
- Escribir: *sudo apt-get install libapache2-mod-auth-mysql*. Con esto instalamos la conexión con apache.
- Escribir: *sudo apt-get install php5-mysql*. Con esto instalamos la conexión con php.
- Escribir: *sudo /etc/init.d/apache2 restart*. Reiniciar el servidor apache.
- Escribir: *sudo /etc/init.d/mysql restart*. Reiniciar mysql.

B.3. Instalando Librerías Necesarias

Para el correcto funcionamiento de la aplicación es necesario instalar dos librerías de Php que no vienen en la instalación por defecto.

La primera es la librería **GD**, encargada del procesamiento de imagenes. La librería GD es necesaria debido a su uso en las gráficas mostradas con la librería PHPlot. La segunda es la librería **SVN**, que recoge todas las funciones de comunicación con un repositorio subversion usadas en la aplicación, además de muchas otras.

B.3.1. Instalar la librería GD

Para instalar esta librería bastará:

- Abrir shell.
- Escribir: *sudo apt-get install php5-gd*.

B.3.2. Instalar la librería SVN

Para instalar esta librería debermos hacerlo a través de un repositorio de extensiones de PHP (PECL):

- Abrir shell.
- Escribir: *sudo apt-get install php-pear*. Con esto podremos acceder al repositorio PECL.
- Escribir: *sudo apt-get install libsvn-dev*.
- Escribir: *sudo pecl install svn*. Instalamos la librería SVN a través del repositorio PECL.
- Para finalizar debemos abrir el archivo *php.ini* y añadir como extensión "svn.so".

B.4. Comprobando el Funcionamiento

Para comprobar el funcionamiento de apache, si estás en la misma máquina donde instalaste Apache, coloca en la barra de direcciones de tu navegador: *http://localhost* o *http://127.0.0.1*.

Si te encuentras en una máquina distinta a la que instalaste Apache, debes colocar en la barra de direcciones de tu navegador: *http://192.168.0.1*. Donde debes reemplazar 192.168.0.1 por la IP de la máquina en la que instalaste Apache. Esto te llevará a una página que nos redireccionará a *http://localhost/apache2-default/* o *http://192.168.0.1/apache2-default/*, dependiendo del caso, la cual nos indicará que el Servidor Apache ha sido instalado correctamente.

B.5. Instalar la Aplicación

Una vez instalado todos los programas necesarios para la ejecución de la aplicación, para usarla simplemente tendremos que llevar a cabo dos tareas más, la primera será crear la base de datos en el sistema, para lo cual se facilita un archivo *.sql* que al ejecutarlo creará todas las tablas necesarias para el funcionamiento de la aplicación. Por otro lado, la segunda tarea que tendremos que realizar será la de alojar la aplicación en la carpeta servidor que hemos creado con anterioridad (*/var/www*).

B.5.1. Creando la Base de Datos

Antes de crear la base de datos, hemos de copiar el archivo *estdv.sql* existente en la carpeta DB del CD-ROM, en una ubicación conocida, ya que lo necesitaremos para generar la base de datos.

Para crear la base de datos, hemos de abrir la consola del sistema. En esta pantalla, hemos de escribir el siguiente comando, *mysql -u root -p <ubicacion_del_archivo_sql* donde "ubicacion_del_archivo_sql" se refiere al lugar del sistema donde está almacenado el archivo *estdv.sql* que copiamos del CD-ROM. Si, por ejemplo, lo hemos ubicado en el escritorio, el comando que a escribir es el siguiente: *mysql -u root -p </home/user/Escritorio/estdv.sql* (user es el usuario del sistema). Tras escribirlo se nos pedirá que introduzcamos la contraseña que de administrador de MySQL. Con esto tendremos la base de datos del sistema creada y lista para funcionar.

B.5.2. Copiando los Archivos

Una vez creada la Base de Datos, sólo falta realizar la copia de todos los archivos de la aplicación a la carpeta servidor (*/var/www*), que creamos al instalar el servidor Apache. Para ello, descomprimos el archivo EstDV y la copiamos en nuestra carpeta servidor. Finalmente tendríamos una carpeta en */var/www/system* y un archivo */var/www/index.php*.

B.5.3. Otorgando Permisos

Tras haber realizado la copia de todos los archivos, sólo falta configurar los permisos, para poner a punto la aplicación y poder iniciarla. Necesitamos alterar los permisos de lectura y escritura de la carpeta donde está la aplicación (*estdv*) para que la aplicación pueda realizar todas sus funciones sin problemas.

Para esto abrimos la consola, y nos dirigimos a la carpeta servidor y damos los permisos a la carpeta *estdv*:

- Abrir shell.
- Escribir: *cd /var/www*. Vamos al directorio servidor.
- Escribir: *sudo chmod 777 estdv*. Damos permisos a la aplicación.
- Escribir: **contraseña**. Nos pedirá la contraseña antes de realizar la acción.

Con esto ya tenemos los permisos adecuados para que la aplicación funcione correctamente.

B.5.4. Iniciando la Aplicación

Para iniciar la aplicación simplemente tendremos que dirigirnos al navegador web, y escribir en la barra de dirección, lo siguiente: *localhost/estdv/index.php*, si la instalación ha sido satisfactoria, se debería de acceder a la página de inicio del sitio web.

Por defecto el usuario y contraseña de Administrador es **Admin** y **1234** respectivamente. Es recomendable cambiar la contraseña lo antes posible.

Anexos

Anexo 1: Contenido del CD-ROM

El CD-ROM adjunto contiene:

- La documentación de la aplicación (este documento) en formato PDF.
- El manual de usuario en formato PDF.
- El manual de instalación en formato PDF.
- Un archivo comprimida, llamada EstDV, que contiene la aplicación.
- Una carpeta, llamada DB, que contiene el archivo .sql para la instalación de la base de datos MySQL.

Bibliografía

- [1] Christopher Murphy & Nicklas Persson. *HTML y CSS*. Anaya Multimedia (Friends of). 2009.
- [2] Julie C. Meloni. *PHP, MYSQL y APACHE*. Anaya Multimedia (SAMS). 2008.
- [3] *Apuntes de la Asignatura Bases de Datos (ITIS)*. 2008-2009.
- [4] *Apuntes de la Asignatura Ingeniería del Software (ITIS)*. 2008-2009.
- [5] Antonio Gámez Mellado, Alejandro Pérez Cuellar, Alicia Cornejo Barrios, Aris Burgos Pintos, Francisco José González Gutiérrez & Luis M. Marín Trechera. *Taller de L^AT_EX*. 2010.
- [6] David Upton. *Codeigniter For Rapid Php Application Development*.
- [7] Glass, Michael. *Desarrollo Web con PHP y MySQL*. Anaya Multimedia. 2004.
- [8] *Foros de PHP, CSS, HTML y MYSQL*.
- [9] <http://php.net/manual/es/index.php>
- [10] <http://php.net/manual/en/book.svn.php>
- [11] <http://dev.mysql.com/doc/refman/5.0/es/index.html>
- [12] <http://www.desarrolloweb.com/manuales/manual-css-hojas-de-estilo.html>
- [13] http://codeigniter.com/user_guide
- [14] <http://www.desarrolloweb.com/manuales/21/>
- [15] <http://www.desarrolloweb.com/manuales/manual-rss.html>
- [16] <http://www.desarrolloweb.com/articulos/1180.php>
- [17] <http://phplot.sourceforge.net/phplotdocs/>
- [18] <http://www.cssblog.es/40-plantillas-css-y-xhtml-de-alta-calidad/>

GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

<<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “**Document**”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “**you**”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “**Opaque**”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors,

SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “**Title Page**” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “**publisher**” means any person or entity that distributes copies of the Document to the public.

A section “**Entitled XYZ**” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “**Acknowledgements**”, “**Dedications**”, “**Endorsements**”, or “**History**”.) To “**Preserve the Title**” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or non-commercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to

the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with ... Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.